

# Universidad de Alcalá

## Escuela Politécnica Superior

### Máster Universitario en Ingeniería de Telecomunicación

#### Trabajo Fin de Máster

Diseño, implementación y evaluación de un sistema de simulación  
acústica en entornos cerrados

ESCUELA POLITECNICA  
SUPERIOR

**Autor:** Ruth Pérez Pascual

**Tutores:** Cristina Losada Gutiérrez y Javier Macías Guarasa

2016



# UNIVERSIDAD DE ALCALÁ

## ESCUELA POLITÉCNICA SUPERIOR

**Máster Universitario en Ingeniería de Telecomunicación**

**Trabajo Fin de Máster**

**Diseño, implementación y evaluación de un sistema de  
simulación acústica en entornos cerrados**

Autora: Ruth Pérez Pascual

Tutores: Cristina Losada Gutiérrez y Javier Macías Guarasa

**Tribunal:**

**Presidente:** Manuel Mazo Quintas

**Vocal 1º:** Manuel Utrilla Manso

**Vocal 2º:** Cristina Losada Gutiérrez

Fecha: 20 de septiembre de 2016



**A los escépticos.**

*“El hombre nunca sabe de lo que es capaz hasta que lo intenta”*  
Charles Dickens



# Resumen

En este Trabajo Fin de Máster se realiza un estudio de las principales herramientas de simulación acústica con el objetivo de diseñar, implementar y evaluar un nuevo sistema de simulación acústica aplicado a entornos cerrados. El sistema asumirá un modelo de propagación de rayos, y el objetivo es obtener las trayectorias seguidas por los rayos acústicos, determinadas a través de las reflexiones especulares y las difracciones en bordes, teniendo en cuenta la geometría de la sala. Como tareas adicionales, se ha realizado un entorno de visualización en 3D de las trayectorias calculadas mediante teoría geométrica, y una API que proporciona acceso a los resultados por parte de otras aplicaciones.

**Palabras clave:** Matlab, simulación acústica, método fuente imagen, difracciones en bordes, recintos cerrados.





# Abstract

In this project, a study of the main acoustic simulation tools is performed in order to design, implement and evaluate a new acoustic simulation system applied to closed environments. The system will assume a ray propagation model, and the objective is to obtain the paths followed by the acoustic rays, determined by specular reflections and edge diffractions, taking into account the geometry of the room. As additional tasks, it has performed a 3D visualization environment of the trajectories calculated by means of geometric theory, and a API that provides access to the results by other applications.

**Keywords:** Matlab, acoustic simulation, image source method, edge diffraction, closed environment.



# Resumen extendido

El campo de la acústica ha sido ampliamente estudiado por múltiples científicos e investigadores a lo largo de los siglos. Es por ello que existen multitud de teorías para modelar la propagación de las ondas acústicas en un espacio determinado, bajo un conjunto de condiciones dadas. Los avances en tecnología han permitido en los últimos tiempos obtener sistemas de simulación acústica cada vez más eficientes. En este Trabajo Fin de Máster se trata de desarrollar un sistema de simulación acústica destinado a recintos cerrados capaz de determinar el campo sonoro del mismo. Más concretamente, el sistema está destinado a conocer la respuesta de un recinto ante una señal de audio evaluándola en distintas posiciones. Con el objetivo de conocer la respuesta del recinto, se realiza un estudio de las distintas teorías acústicas que se pueden aplicar al sistema de simulación, seleccionando aquella que más ventajas aporta a este caso concreto. Cabe destacar el uso de una teoría geométrica con el fin de modelar la onda sonora como un rayo, aplicando el método fuente imagen perteneciente a esta teoría.

Un esfuerzo importante se realiza en la investigación de las distintas herramientas de simulación existentes, con especial interés por aquellas con licencia de software libre con el objetivo de disponer del código fuente de la misma. Al resultar una de las herramientas válida para la generación del sistema de simulación a realizar, es estudiada en profundidad y utilizada en el sistema implementado.

El sistema de simulación desarrollado está formado por una interfaz gráfica de usuario realizada a través del entorno de desarrollo GUI de Matlab que permite la utilización del sistema por usuarios con pocos conocimientos en programación. Además, se desarrolla una interfaz de programación de aplicaciones en lenguaje de programación de Matlab que facilite la integración de este sistema de simulación en cualquier otra herramienta donde sea requerido. En el primer caso los resultados son mostrados en un entorno de visualización 3D donde es posible comprobar gráficamente la propagación de los rayos sonoros en el recinto, según la posición de la fuente y receptor. Además, es posible seleccionar en esta interfaz aquellos trayectos que contienen un fenómeno concreto, bien sea reflexión especular o difracción en bordes, aquellos con un orden de propagación concreto, o algunas otras opciones de representación de información calculada. Por su parte, en la interfaz de programación los resultados de la simulación son almacenados en una estructura donde cada matriz contiene los trayectos calculados para cada par fuente-receptor indicado en la simulación.

El sistema desarrollado es capaz de simular recintos con cualquier geometría, obteniendo las trayectorias de los rayos propagados en un recinto cerrado, teniendo en cuenta el fenómeno de reflexión especular y difracción en bordes. Se permite determinar la posición de las fuentes y receptores, así como el orden máximo de las trayectorias calculadas. También es posible el cálculo de la respuesta al impulso del recinto. La geometría del recinto se puede definir a partir de un formato `.env/.srf` o `.cad`, este último generado a partir del software CATT-Acoustics.



# Índice general

Resumen	vii
Abstract	ix
Resumen extendido	xi
Índice general	xiii
Índice de figuras	xvii
Índice de tablas	xix
Lista de acrónimos	xxii
<b>1 Introducción</b>	<b>1</b>
1.1 Motivación y objetivos . . . . .	2
1.2 Organización de la memoria . . . . .	3
<b>2 Fundamentos teóricos</b>	<b>5</b>
2.1 Introducción . . . . .	5
2.2 Onda sonora . . . . .	5
2.2.1 Velocidad de propagación del sonido . . . . .	5
2.2.2 Ecuación de onda . . . . .	6
2.2.3 Parámetros de la onda sonora . . . . .	7
2.3 Propagación del sonido . . . . .	8
2.3.1 Sonido directo . . . . .	9
2.3.2 Reflexión . . . . .	9
2.3.3 Difracción . . . . .	11
2.3.4 Refracción . . . . .	12
2.4 Teorías acústicas . . . . .	13
2.4.1 Teoría estadística . . . . .	14
2.4.2 Teoría geométrica . . . . .	15

2.4.2.1	Método fuente-imagen . . . . .	15
2.4.2.2	Método de trazado de rayos . . . . .	19
2.4.2.3	Método de trazado de haz . . . . .	21
2.4.2.4	Método de trazado de conos . . . . .	21
2.4.2.5	Comparativa . . . . .	21
2.4.3	Teoría ondulatoria . . . . .	22
2.4.3.1	Método de elementos finitos . . . . .	22
2.4.3.2	Método de elementos de contorno . . . . .	23
2.4.3.3	FDTD . . . . .	23
2.4.3.4	Comparativa . . . . .	23
2.4.4	Comparativa . . . . .	24
2.5	Parámetros acústicos . . . . .	25
2.5.1	Coefficiente de absorción sonora . . . . .	25
2.5.2	Tiempo de reverberación . . . . .	25
2.5.3	Claridad . . . . .	26
2.6	Conclusiones . . . . .	26
<b>3</b>	<b>Evaluación de sistemas de simulación</b>	<b>29</b>
3.1	Introducción . . . . .	29
3.2	Sistemas comerciales . . . . .	29
3.2.1	Breve descripción . . . . .	29
3.2.2	Comparativa . . . . .	31
3.3	Sistemas libres . . . . .	32
3.3.1	Breve descripción . . . . .	32
3.3.2	Comparativa . . . . .	35
3.3.3	Selección . . . . .	36
3.4	Edge Diffraction Toolbox . . . . .	36
3.4.1	Autor . . . . .	38
3.4.2	Licencia . . . . .	38
3.4.3	Versiones de la herramienta . . . . .	38
3.4.4	EDT versión B2 . . . . .	38
3.4.4.1	Descripción . . . . .	38
3.4.4.2	Método de cálculo . . . . .	39
3.4.4.3	Tiempo de cálculo y uso de memoria . . . . .	42
3.5	Otras herramientas de interés . . . . .	42
3.6	Conclusiones . . . . .	43

<b>4</b>	<b>Desarrollo</b>	<b>45</b>
4.1	Introducción	45
4.2	Objetivos del sistema de simulación	46
4.3	Preprocesamiento	46
4.3.1	Interfaz gráfica	46
4.3.1.1	Conversión del formato de archivos de la geometría	46
4.3.1.1.1	env2matlab(ruta,envfile):	48
4.3.1.1.2	matlab2cad():	48
4.3.1.2	Creación del archivo de configuración	50
4.3.1.3	Representación del modelo	50
4.3.2	Interfaz de programación	50
4.4	Simulación	50
4.4.1	Interfaz gráfica	51
4.4.2	Interfaz de programación	51
4.5	Postprocesamiento	51
4.5.1	Interfaz gráfica	52
4.5.2	Interfaz de programación	54
4.6	Aspectos a considerar en el sistema de simulación	54
4.7	Conclusiones	56
<b>5</b>	<b>Evaluación y resultados</b>	<b>57</b>
5.1	Introducción	57
5.2	IdiapRoom	57
5.2.1	Caso 1	57
5.2.1.1	Reflexión y difracción de orden 2	59
5.2.1.2	Reflexión de orden 6	59
5.2.2	Caso 2	59
5.2.2.1	Reflexión y difracción de orden 2	59
5.2.2.2	Reflexión de orden 8	61
5.3	demoRoom3	61
5.3.1	Caso 1	62
5.3.1.1	Reflexión y difracción de orden 2	63
5.3.1.2	Reflexión de orden 9	63
5.4	Ispace	63
5.4.1	Caso 1	65
5.4.1.1	Reflexión y difracción de orden 2	65
5.4.1.2	Reflexión de orden 5	65

5.4.2	Caso 2 . . . . .	67
5.4.2.1	Reflexión y difracción de orden 2 . . . . .	67
5.4.2.2	Reflexión de orden 5 . . . . .	67
5.4.3	Caso 3 . . . . .	69
5.4.3.1	Reflexión y difracción de orden 2 . . . . .	70
5.4.3.2	Orden 5 de reflexión . . . . .	70
5.5	Conclusiones . . . . .	70
<b>6</b>	<b>Conclusiones y líneas futuras</b>	<b>73</b>
6.1	Introducción . . . . .	73
6.2	Conclusiones . . . . .	73
6.3	Líneas futuras . . . . .	74
<b>7</b>	<b>Presupuesto</b>	<b>75</b>
7.1	Costes de equipamiento . . . . .	75
7.2	Costes de mano de obra . . . . .	75
7.3	Coste total del presupuesto . . . . .	76
	<b>Bibliografía</b>	<b>77</b>
<b>A</b>	<b>Licencias de software</b>	<b>83</b>
A.1	Software propietario . . . . .	83
A.2	Software libre . . . . .	84
<b>B</b>	<b>Manual de usuario</b>	<b>87</b>
B.1	Introducción . . . . .	87
B.2	Manual de la interfaz gráfica . . . . .	87
B.2.1	Interfaz de bienvenida . . . . .	87
B.2.2	Interfaz de introducción de parámetros . . . . .	87
B.2.2.1	Archivos . . . . .	89
B.2.2.2	Fuentes y receptores . . . . .	89
B.2.2.3	Opciones de simulación . . . . .	90
B.2.3	Interfaz de visualización de resultados . . . . .	92
B.3	Manual de la interfaz de programación . . . . .	94



# Índice de figuras

2.1	Frentes de onda de una onda plana. (Fuente: basada en [1]) . . . . .	7
2.2	Frentes de onda de una onda esférica. (Fuente: basada en [1]) . . . . .	7
2.3	Representación del nivel sonoro en un recinto. (Fuente: basada en [2]) . . . . .	9
2.4	Sonido directo . . . . .	9
2.5	Reflexiones en un recinto cerrado . . . . .	10
2.6	Tipo de reflexión en función de la frecuencia. (Fuente: basada en [2]) . . . . .	11
2.7	Fenómeno de difracción en un obstáculo . . . . .	11
2.8	Fenómeno de refracción en un recinto con sistema de acondicionamiento de temperatura . . . . .	13
2.9	Esquema del método fuente-imagen . . . . .	16
2.10	Aplicación método fuente-imagen . . . . .	17
2.11	Posición de VS. (Fuente: basado en [1]) . . . . .	17
2.12	Comprobación de visibilidad entre VS y oyente. (Fuente: basado en [1]) . . . . .	18
2.13	Pasos principales del método FEM . . . . .	23
2.14	División del espectro audible en zonas . . . . .	24
2.15	Variación de la densidad de energía sonora al emitir un sonido continuo . . . . .	26
3.1	Esquema de cálculo de la herramienta EDT . . . . .	39
4.1	Pantalla de bienvenida de la interfaz gráfica de usuario. . . . .	47
4.2	Pantalla de introducción de parámetros de la interfaz gráfica de usuario. . . . .	47
4.3	Ejemplo de sentido de las normales en un recinto con obstáculos . . . . .	49
4.4	Pantalla visualización de resultados de la interfaz gráfica de usuario. . . . .	52
5.1	Recinto IdiapRoom . . . . .	58
5.2	Posición fuente y receptor caso 1 Idiaprooom . . . . .	58
5.3	Simulación IdiapRoom con orden 2 de reflexión y orden 1 de difracción . . . . .	59
5.4	Tipos de difracciones en IdiapRoom . . . . .	59
5.5	Simulación IdiapRoom con orden 6 de reflexión . . . . .	60
5.6	Posición fuente y receptor caso 2 IdiapRoom . . . . .	60
5.7	Simulación IdiapRoom caso 2 con orden 2 de reflexión y orden 2 de difracción . . . . .	61

5.8 Simulación IdiapRoom caso 2 con orden 8 de reflexión . . . . .	61
5.9 Recinto demoRoom3 . . . . .	62
5.10 Posición fuente y receptor demoRoom3 . . . . .	62
5.11 Simulación demoRoom3 con orden 2 de reflexión y difracción . . . . .	63
5.12 Simulación demoRoom3 con orden 9 de reflexión . . . . .	64
5.13 Recinto Ispace . . . . .	64
5.14 Posición fuente y receptor caso 1 Ispace . . . . .	65
5.15 Simulación Ispace caso 1 con orden 2 de reflexión y difracción . . . . .	66
5.16 Detalle trayectoria Ispace con dos órdenes de difracción . . . . .	66
5.17 Simulación Ispace caso 1 con orden 5 de reflexión . . . . .	67
5.18 Posición fuente y receptor caso 2 Ispace . . . . .	68
5.19 Simulación Ispace caso 2 con orden 2 de reflexión y difracción . . . . .	68
5.20 Simulación Ispace caso 2 con orden 5 de reflexión . . . . .	69
5.21 Posición fuente y receptor caso 3 Ispace . . . . .	69
5.22 Simulación Ispace caso 3 con orden 2 de reflexión y difracción . . . . .	70
5.23 Simulación Ispace caso 3 con orden 5 de reflexión . . . . .	71
 A.1 Mapa conceptual del software libre (Fuente: es.gnu.org) . . . . .	 85
 B.1 Pantalla de bienvenida al interfaz gráfico . . . . .	 88
B.2 Pantalla de introducción de parámetros del interfaz gráfico . . . . .	88
B.3 Ampliación de la interfaz de introducción de parámetros relativos a archivos. . . . .	89
B.4 Ampliación de la interfaz de introducción de parámetros relativos a fuentes y receptores. . . . .	90
B.5 Ampliación de la interfaz de introducción de parámetros relativos a las opciones de simulación. . . . .	91
B.6 Pantalla de visualización de la interfaz gráfica . . . . .	92
B.7 Ampliación de la interfaz de visualización de resultados . . . . .	92
B.8 Ampliación de la interfaz de visualización de resultados . . . . .	93
B.9 Ampliación de la interfaz de visualización de resultados . . . . .	93
B.10 Ampliación de la interfaz de visualización de resultados . . . . .	94

# Índice de tablas

3.1	Comparativa funcional de sistemas de simulación comerciales . . . . .	31
3.2	Comparativa de sistemas de simulación con licencia de software propietario . . . . .	31
3.3	Características herramientas de simulación con licencia libre . . . . .	37
5.1	Posición fuente y receptor caso 1 Idiaprooom . . . . .	58
5.2	Posición fuente y receptor caso 2 Idiaprooom . . . . .	60
5.3	Posición fuente y receptor demoRoom3 . . . . .	62
5.4	Posición fuente y receptor caso 1 Ispace . . . . .	65
5.5	Posición fuente y receptor caso 2 Ispace . . . . .	67
5.6	Posición fuente y receptor caso 3 Ispace . . . . .	70
7.1	Coste del equipamiento hardware utilizado . . . . .	75
7.2	Coste del equipamiento software utilizado . . . . .	75
7.3	Coste de mano de obra . . . . .	75
7.4	Coste total del presupuesto . . . . .	76
B.1	Ejemplos de valores en matriz raytype . . . . .	95
B.2	Ejemplos de valores en matriz paths . . . . .	96



# Lista de acrónimos

AGPL     *Affero General Public License.*  
API     *Application Programming Interface.*

BEM     *Boundary Element Method.*  
BSD     *Berkeley Software Distribution.*  
BTM     *Biot-Tolstoy-Medwin.*

CeCILL   *CEA CNRS INRIA Logiciel Libre.*

ECL     *Educational Community License.*  
EDT     *Edge Diffraction Toolbox.*  
EPL     *Eclipse Public License.*

FD     *Finite-Difference.*  
FDTD   *Finite-Difference Time-Domain.*  
FEM     *Finite Element Method.*

GNU     *GNU's Not Unix.*  
GPL     *General Public License.*  
GUI     *Graphical User Interface.*

IR     *Impulsive Response.*  
ISM     *Image-Source Method.*

KZK     *Khokhlov-Zabolotskaya-Kuznetsov.*

LEM     *Lumped Element Method.*  
LGPL   *Lesser General Public License.*

MIT     *Massachusetts Institute of Technology.*  
MLS     *Maximum Length Sequence.*  
MPI     *Message Passing Interface.*  
MPL     *Mozilla Public License.*

PML	<i>Perfectly Matched Layers.</i>
PSTD	<i>Pseudo-Spectral Time-Domain.</i>
RMS	<i>Root Mean Square.</i>
SEA	<i>Statical Energy Analysis.</i>
TF	<i>Transfer Function.</i>
TFM	Trabajo Fin de Máster.
VS	<i>Virtual Source.</i>

# Capítulo 1

## Introducción

La acústica [3] es la rama de la física encargada del estudio de la producción, transmisión, almacenamiento, percepción y reproducción del sonido, infrasonido y ultrasonido. Se entiende por sonido aquella perturbación que se propaga en forma de onda sonora a través de un medio elástico produciendo variaciones de presión y vibraciones de las partículas que pueden ser percibidas por el oído humano u otro instrumento. El oído humano es capaz de percibir ondas entre  $20\text{ Hz}$  y  $20\text{ kHz}$ , siendo considerada esta franja del espectro audible como sonido, mientras que se denomina infrasonido a frecuencias por debajo de  $20\text{ Hz}$  y ultrasonido a frecuencias por encima de  $20\text{ kHz}$ . En este Trabajo Fin de Máster (TFM) se centra la atención en los fenómenos acústicos que sufren las ondas sonoras al ser propagadas en un recinto cerrado.

El campo de la acústica es muy versátil, por este motivo ha sido aplicado en multitud de áreas, especialmente y con interés en este TFM en aquellas en las que interviene un entorno o recinto y la transmisión de las ondas acústicas a través del mismo. De este modo se encuentra la arqueoacústica, encargada del estudio de los efectos acústicos que se producen en lugares arqueológicos, o la acústica arquitectónica [4], que se encarga del estudio del aislamiento sonoro entre recintos colindantes y del acondicionamiento acústico de recintos en los que se reproducirá el sonido. Especial interés en este caso tiene la acústica arquitectónica ya que estudia el control acústico en recintos abiertos o cerrados.

Datan de la Antigüedad los primeros escritos conocidos sobre acústica arquitectónica, enfocados principalmente a la generación y transmisión del sonido en recintos típicos de la época, como teatros o iglesias. En aquella época los diseños de recintos realizados tenían una componente práctica que consistía en imitar colocaciones de salas o entornos ya existentes donde la música sonaba bien. No fue hasta finales del siglo XIX el nacimiento de la conocida como acústica arquitectónica moderna de la mano del famoso físico estadounidense Wallace Clement Sabine. La principal aportación de Sabine a la acústica arquitectónica tiene que ver con la reverberación, objeto de estudio en cualquier diseño de acústica arquitectónica hoy en día por su gran impacto en la calidad sonora de un recinto.

En este TFM se genera un sistema de simulación acústica para aplicación en recintos cerrados, centrando el esfuerzo en la determinación de la propagación de los rayos sonoros desde que son emitidos por la fuente sonora hasta que llegan a un determinado oyente o receptor. Los avances en tecnología han permitido desarrollar sistemas de simulación acústica cada vez más eficientes, desarrollando diversas técnicas de simulación para determinar y analizar la propagación del sonido, en un espacio determinado, bajo un conjunto de condiciones dadas. Estas técnicas tienen múltiples aplicaciones en diferentes ámbitos como el diseño del interior de vehículos o altavoces. Existen múltiples métodos para la simulación acústica en entornos cerrados, cada uno de los cuales con una serie de ventajas e inconvenientes. Por un lado se

encuentra la teoría ondulatoria que resuelve la ecuación de onda numéricamente aportando resultados muy precisos pero con tiempos de procesamiento muy elevados y por otro lado, la teoría geométrica basada en la representación de las ondas acústicas como rayos. Cada uno de los métodos dentro de cada una de estas teorías tiene sus propias características y aplicaciones, por tanto, es necesario el estudio y análisis de cual supone la mejor opción para un sistema de simulación concreto.

Centrando la atención en la teoría geométrica por ser la aplicada en el sistema de simulación desarrollado, cabe destacar que según las condiciones del recinto, además de otros parámetros se elige un método u otro. Por ejemplo, si nos encontramos ante un recinto abierto el fenómeno predominante es el de difusión sonora. Sin embargo, si nos encontramos ante un recinto cerrado predomina el fenómeno de la reflexión. En el primer caso, sería más conveniente la aplicación del método de trazado de rayos ya que es capaz de determinar las reflexiones difusas, acercándose de este modo más a las condiciones del recinto. En el segundo caso, ante un recinto cerrado, funcionarían prácticamente igual tanto el método de fuente imagen como el de trazado de rayos, teniendo que realizar la elección del método a aplicar por otra condición, por ejemplo, la exactitud de resultados obtenidos.

De modo genérico cabe destacar que en la simulación acústica de recintos se distinguen varias fases, a saber: antes de comenzar con la simulación es necesario conocer la geometría del recinto donde se van a propagar las ondas sonoras o rayos sonoros, así como la ubicación de las fuentes sonoras y de los receptores. En segundo lugar, mediante alguno de los métodos de las distintas teorías de simulación acústica, se determinan las componentes (ondas o rayos) que perciben los distintos receptores especificados. En base a este conocimiento es posible calcular las distintas respuestas al impulso (*Impulsive Response* (IR)) generando una respuesta al impulso total del recinto. Esta respuesta total permite calcular un conjunto de parámetros que aportan unas medidas de calidad acústica para definir al recinto bajo estudio, además de otros aspectos.

En resumen, en la actualidad son múltiples los ámbitos en los que se incluye la simulación acústica por diversos motivos, desde la popular simulación acústica de recintos para determinar la calidad sonora del mismo hasta el reciente fenómeno de auralización [5]. Este último concepto de la acústica, con cada vez más estudios entorno a él, fue introducido por primera vez en 1990 y se utiliza como complemento de los tradicionales parámetros acústicos, analizando la calidad acústica de una sala. A grandes rasgos, se trata del proceso mediante el que se simula la sensación de escuchar sonido tridimensional en el interior de un recinto, conociendo la posición de las fuentes sonoras y de los oyentes.

## 1.1 Motivación y objetivos

La motivación de este TFM surge de la línea de procesamiento de audio basado en agrupaciones de micrófonos que aborda el grupo de investigación GEINTRA de la Escuela Politécnica Superior de la UAH. En varias de las aplicaciones de trabajo del grupo en esa línea se hacía imprescindible contar con una herramienta de simulación acústica que permitiera conocer la respuesta de un recinto ante una señal de audio (correspondiente, en general, a la señal de habla que genera un usuario dentro de dicho recinto), evaluándola en posiciones determinadas del espacio (donde estarían situados micrófonos).

Las aplicaciones iniciales en las que se integrará el trabajo desarrollado son la de generación de un modelo predictivo de la energía acústica generada por fuentes de audio en entornos cerrados [6–8], y la de posicionamiento óptimo de agrupaciones de micrófonos [9].

El objetivo general de este TFM es el diseño, implementación y evaluación de un sistema de simulación acústica en entornos cerrados. Por tanto, los objetivos principales de éste son:



1. Estudio de la teoría acústica, incidiendo en sus principales teorías y los distintos métodos que componen cada una de ellas.
2. Realización de un análisis exhaustivo del estado del arte en sistemas de simulación acústica, evaluando las alternativas disponibles en la literatura, y analizando sus ventajas e inconvenientes en relación a las características necesarias en el trabajo.
3. Evaluación de la posibilidad de modificación de alguno de los sistemas analizados, requiriendo que sea de código abierto, disponga de literatura de cara a adecuarlo a las restricciones del problema planteado, o realizar la implementación completa del sistema.
4. Diseño e implementación de un sistema de simulación acústica en entornos cerrados que cumpla los siguientes requisitos:
  - (a) Permitir la definición y modificación sencilla de la geometría del entorno acústico que se desea simular, así como la incorporación de obstáculos.
  - (b) Permitir la definición de las fuentes sonoras y los elementos de captura de las mismas.
  - (c) Ser flexible para permitir modificaciones, bien sea en los parámetros de simulación o para adecuar el software a ámbitos similares.
5. Evaluación de los algoritmos implementados, realizando experimentos con las bases de datos disponibles, para comprobar el cumplimiento de todos los requisitos especificados.
6. Documentación del trabajo realizado.

## 1.2 Organización de la memoria

Esta memoria está organizada en siete capítulos y dos anexos, los cuales se describen a continuación:

1. Introducción: en el Capítulo 1 se trata de dar una visión global del trabajo realizado, situándolo dentro de su contexto temático. Además, se especifica la motivación y los objetivos del mismo.
2. Fundamentos teóricos: en el Capítulo 2 se realiza un repaso de la teoría acústica desde los principios de la ecuación de onda hasta la descripción de los distintos métodos de simulación acústica. También se realiza un breve repaso de algunos parámetros acústicos relevantes.
3. Evaluación de sistemas de simulación: en el Capítulo 3 se realiza un estudio del estado del arte de distintos sistemas de simulación acústica existentes tanto comerciales como libres. Se especifican distintos aspectos de los mismos con el objetivo de llevar a cabo una comparación y de este modo seleccionar aquel que pueda ser de utilidad para el desarrollo del sistema de simulación a implementar.
4. Desarrollo: en el Capítulo 4 se muestran los distintos desarrollos y modificaciones llevados a cabo para realizar el sistema de simulación resultante.
5. Evaluación y resultados: en el Capítulo 5 se muestran algunos ejemplos de funcionamiento del sistema de simulación desarrollado, pudiendo observar los resultados arrojados por el programa.
6. Conclusiones y líneas futuras: en el Capítulo 6 se establecen una serie de conclusiones acerca del diseño, evaluación e implementación del sistema de simulación acústica. Además, se proponen una serie de trabajos futuros en la línea de este TFM.

7. Presupuesto: en el Capítulo 7 se realiza un cálculo del coste necesario para la realización de este TFM.
8. Apéndices:
  - A Licencias software: En el Apéndice A se enumeran los principales tipos de licencias software, haciendo mayor hincapié en las licencias libres para entender la selección realizada en el Capítulo 3.
  - B Manual de usuario: en el Apéndice B se explican los pasos necesarios para el uso del sistema de simulación desarrollado, así como la utilidad de las distintas opciones del mismo.

# Capítulo 2

## Fundamentos teóricos

### 2.1 Introducción

En este capítulo se realiza un repaso de los fundamentos que rigen la teoría acústica de modo que se construya una base que permita el correcto entendimiento del presente trabajo.

En primer lugar se describe brevemente el elemento principal de la acústica, la onda sonora. Se realiza una definición y se especifican algunos de los parámetros más relevantes que la caracterizan. En segundo lugar, se describen las leyes físicas que rigen la propagación del sonido y que son de utilidad a la hora de generar el sistema de simulación acústica del trabajo. El siguiente apartado trata los distintos tipos de teorías acústicas existentes para simulación de recintos, destacando en cada una de ellas algunos de los métodos más importantes, haciendo hincapié en aquel que se aplica al sistema de simulación desarrollado. Por último, se realiza un breve repaso de los parámetros acústicos que aportan medidas de calidad del recinto, que aunque en el sistema desarrollado no son calculados, es de utilidad conocerlos por su relevancia en otros tipos de sistemas de simulación acústicos.

### 2.2 Onda sonora

Comenzando por lo más básico que forma un sistema acústico encontramos la onda sonora. Si esta se propaga en un medio elástico y continuo produce variaciones locales de presión que transforman la onda longitudinal sonora inicial en una onda esférica. Esta onda, además de producir variaciones en la presión del medio, se ve influida por la humedad o la temperatura, entre otros factores, que alteran el efecto acústico que el ser humano percibe por el sistema auditivo, conocido como sonido.

#### 2.2.1 Velocidad de propagación del sonido

Según se enuncia en [10], la velocidad de propagación del sonido es constante en espacio y tiempo si se está ante un medio homogéneo, sin pérdidas y sin límites en cualquier dirección. En este caso, se define la velocidad de propagación del sonido como:

$$c = \sqrt{\frac{E}{\rho}} \quad \left[ \frac{m}{s} \right], \quad (2.1)$$

donde  $E$  es el coeficiente de elasticidad en  $N/m^2$  y  $\rho$  es función de la densidad en  $kg/m^3$ .

Como bien se ha adelantado, la velocidad del sonido depende de la temperatura y es por eso que se incluye este parámetro en la ecuación de la velocidad del sonido, según se muestra a continuación:

$$c = 331,4 \sqrt{1 + \frac{\Theta}{273}} \quad \left[ \frac{m}{s} \right], \quad (2.2)$$

donde  $\Theta$  es la temperatura en  $^{\circ}\text{C}$ . Así, en condiciones normales ( $20^{\circ}\text{C}$ , 50 % de humedad y a nivel del mar), la velocidad de propagación del sonido es de  $343 \text{ m/s}$ .

## 2.2.2 Ecuación de onda

Se ha definido la onda sonora como aquella que produce variaciones de presión en un medio, dando lugar al sonido. Estas variaciones de presión vienen motivadas por el continuo movimiento de las partículas que componen la onda. Definiendo la onda por los desplazamientos instantáneos o la velocidad de sus partículas, se puede observar como se crean zonas en el medio con mucha densidad de partículas y otras zonas con menor densidad, produciendo este hecho las variaciones de presión que definen a la onda sonora.

Además de variaciones en la presión, pueden existir en el medio variaciones de temperatura, siendo posible conectar éstos dos parámetros por medio de leyes acústicas. De este modo, se tiene una ecuación diferencial conocida como ecuación de onda (véase ecuación 2.3), que define la propagación del sonido en una, dos y tres dimensiones, en cualquier fluido sin pérdidas según se describe en [11].

$$c^2 \Delta p = \frac{\partial^2 p}{\partial t^2} \quad (2.3)$$

siendo  $p$  la presión sonora,  $t$  el tiempo y  $c$  la velocidad de propagación del sonido con la siguiente forma:

$$c^2 = k \frac{p_0}{\rho_0} \quad (2.4)$$

En el caso de estar ante un recinto paralelepípedo de dimensiones  $x, y, z$  con superficies rígidas y sin absorción, la ecuación de onda tiene la siguiente forma:

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + \frac{\partial^2 p}{\partial z^2} = \frac{\rho_0}{k} \frac{\partial^2 p}{\partial t^2} = \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2} \quad (2.5)$$

siendo:

- $p$  la presión sonora en  $\text{Pa}$ .
- $x, y, z$  las coordenadas cartesianas de propagación en  $m$ .
- $\rho_0$  la densidad volumétrica del fluido.
- $k$  la rigidez volumétrica.
- $c$  la velocidad de propagación del sonido.

Si las magnitudes acústicas dependen únicamente del tiempo y de una única dirección, se puede escribir la ecuación 2.5 de forma simplificada para la dirección  $x$  como:

$$\frac{\partial^2 p}{\partial x^2} = \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2} \quad (2.6)$$

La ecuación diferencial 2.3 tiene una solución directa en una dimensión que es la conocida como onda plana y otra solución obtenida tras transformar la ecuación a coordenadas polares conocida como onda

esférica. A modo de aclaración, se muestra en la Figura 2.1 un esquema de una onda plana en la dirección  $x$  y en la Figura 2.2 un esquema de la onda esférica.

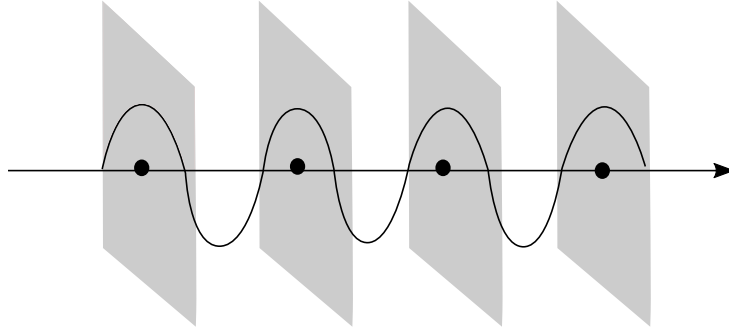


Figura 2.1: Frentes de onda de una onda plana. (Fuente: basada en [1])

La onda plana tiene una presión constante en todo plano perpendicular al eje  $x$ , siendo conocidos estos planos de presión como frentes de onda. Además, cabe destacar que la ecuación que define a la onda plana coincide con la ecuación 2.6 definida anteriormente. En las ondas esféricas, los frentes de onda son esferas concéntricas donde la fuente sonora se encontraría en el centro de la esfera como se puede comprobar esquemáticamente en 2D en la Figura 2.2.

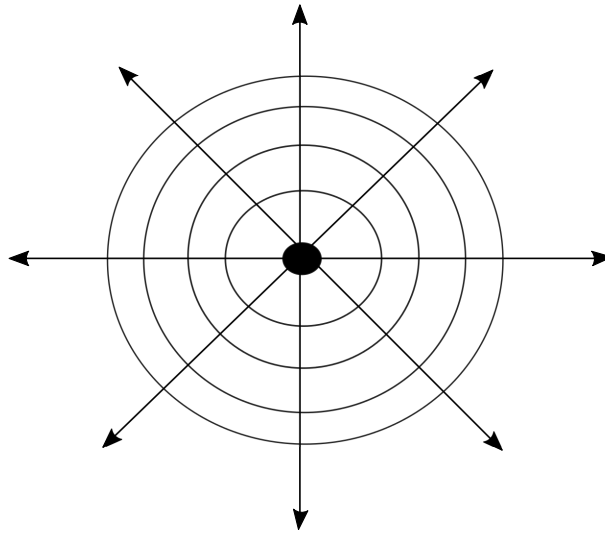


Figura 2.2: Frentes de onda de una onda esférica. (Fuente: basada en [1])

### 2.2.3 Parámetros de la onda sonora

Algunos parámetros que definen a la onda sonora son:

1. **Energía y densidad de energía:** Una onda sonora al propagarse pierde parte de su energía según se puede observar en la ecuación 2.7, donde  $T$  es el periodo de observación de la señal,  $p$  la presión sonora, y  $t$  el tiempo [11].

$$E = \int_0^T p^2(t) dt \quad (2.7)$$

La densidad de energía se caracteriza por ser la energía contenida en una unidad de volumen.

2. **Nivel de presión sonora:** El nivel de presión sonora, también conocido como nivel sonoro, es la relación entre la presión sonora efectiva con respecto a un valor de referencia, según se observa en la ecuación 2.8.

$$L = 10 \log_{10} \left( \frac{\bar{p}^2}{p_0^2} \right) = 20 \log_{10} \left( \frac{\bar{p}}{p_0} \right) [dB] \quad (2.8)$$

En la ecuación 2.8,  $\bar{p}$  es la presión sonora efectiva, también conocida como *Root Mean Square* (RMS), y  $p_0$  es la referencia de presión sonora efectiva en el aire con un valor de  $20 \mu Pa$  que se considera el umbral del oído humano a  $1 kHz$ . En el campo de la acústica, en lugar de hablar de presión en la Unidad Internacional de Unidades ( $Pa$ ), se habla de decibelios ( $dB$ ) ya que de este modo es más fácil aproximarse a la respuesta del oído humano y el amplio rango de presiones sonoras es más manejable. Así, el rango de audición del oído humano se encuentra aproximadamente entre los 0 dB y 130 dB.

3. **Intensidad sonora:** La intensidad sonora es la cantidad de energía que transporta una onda sonora, más concretamente, es la medida de energía que atraviesa una sección plana de  $1 m^2$  y que es perpendicular a la dirección de propagación. Esta intensidad se expresa como el producto de la presión sonora y la velocidad de la partícula.

## 2.3 Propagación del sonido

El estudio de la generación, propagación y recepción del sonido se rige por una serie de leyes de la física acústica. La propagación del sonido modela la forma en la que se propagan las ondas sonoras a través de un medio [12]. Existen una serie de fenómenos que se pueden producir en la propagación del sonido y que se presentaran en este apartado. Además, cabe destacar que existen una serie de técnicas que predicen la trayectoria o recorrido de las ondas acústicas y que se detallan en el apartado 2.4.

Tomando al sonido como una onda esférica y al incidir esta en una superficie no rígida es complejo determinar la reflexión producida. Una superficie no rígida absorbe parte de la energía sonora y otra parte la refleja. No obstante, se asume en este documento que todas las superficies son rígidas y por tanto, reflejan toda la energía que reciben. Aunque el sonido es una onda esférica, si la fuente sonora se encuentra suficientemente alejada de la superficie de reflexión, se puede despreciar la curvatura del frente de onda y considerar ondas planas. Por tanto, es posible partir de un modelo acústico en el que disponemos de superficies totalmente rígidas y ondas planas.

Los fenómenos que pueden producirse en la propagación del sonido son:

- Sonido directo
- Reflexiones
- Difracciones
- Refracciones

Cabe destacar que en el presente documento se está analizando un recinto cerrado en el que se estudiarán los tres primeros fenómenos de la propagación del sonido, a saber: sonido directo, reflexiones y difracciones. Es posible establecer una clasificación de estos fenómenos desde diversos puntos de vista, el primero de ellos es función de si el sonido llega al oyente de forma directa (sonido directo) o de forma indirecta (reflexiones y difracciones). Otro punto de vista es la clasificación dentro del sonido indirecto en función

del tiempo que transcurre desde que el sonido fue emitido y es recibido por el oyente. Se distinguen así las reflexiones tempranas y reflexiones tardías (reverberaciones).

En la Figura 2.3 se observa una relación temporal de las distintas señales que son recibidas por un oyente dentro de un entorno cerrado al excitar un impulso sonoro. La primera de ellas (color verde) se trata del sonido directo con un nivel de intensidad mayor. En azul se observan varias señales que componen las reflexiones tempranas. Por último, con una diferencia de 100 *ms* desde que se emitió el sonido directo, se observan las reverberaciones o reflexiones tardías. Las reflexiones tempranas llegan de forma más espaciada entre sí que las tardías porque los rayos han sufrido menor número de reflexiones en las superficies del recinto y su orden es menor. A medida que avanza el tiempo en una situación real, disminuye la intensidad de las ondas sonoras ya que las continuas reflexiones hacen que pierda energía al absorber las superficies parte de la misma. En el apartado 2.5 se explicará más detenidamente la importancia de la reverberación en la acústica de un recinto.



Figura 2.3: Representación del nivel sonoro en un recinto. (Fuente: basada en [2])

### 2.3.1 Sonido directo

Como se representa en la Figura 2.4 de forma visual, el sonido directo se trata de aquella energía sonora que llega desde la fuente sonora al oyente en línea recta y sin tocar ningún obstáculo o superficie.

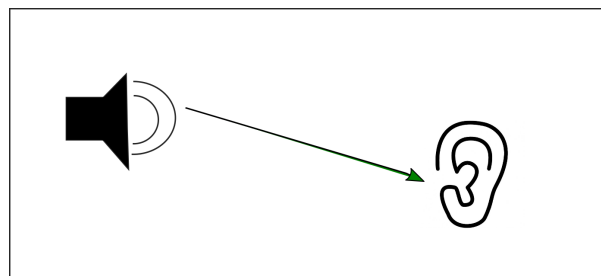


Figura 2.4: Sonido directo

### 2.3.2 Reflexión

La reflexión [13] es el fenómeno producido cuando parte de la energía de la onda sonora incidente se encuentra con un obstáculo o superficie que no puede traspasar ni bordear y es devuelta al medio del que

proviene. En la Figura 2.5 se muestra un ejemplo genérico de este fenómeno.

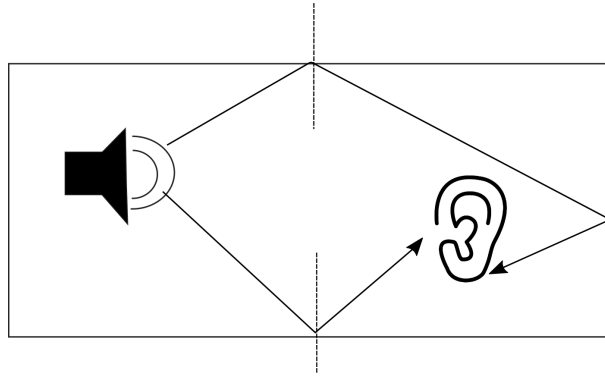


Figura 2.5: Reflexiones en un recinto cerrado

En la referencia [10] se especifica que la cantidad de rayos reflejados que alcanzan al oyente en un tiempo  $t$ , en un recinto de volumen  $V$ , es igual a:

$$N = \frac{4\pi c^3 t^3}{3V} \quad (2.9)$$

En la ecuación 2.9 se puede comprobar como la dependencia cúbica del tiempo provoca que la densidad de rayos no sea uniforme, al comienzo de que la fuente emita la onda sonora, la densidad de reflexiones será baja y la energía será alta; cuando avanza el tiempo, la densidad de reflexiones aumenta y la energía disminuye. Llegará un instante de tiempo en el que no se distinguirán dos reflexiones contiguas y es este fenómeno el que se conoce como reflexiones tardías.

Dentro del fenómeno de la reflexión, es posible distinguir dos tipos, los cuales se detallan a continuación:

1. **Reflexión difusa:** se produce cuando la onda sonora incidente lo hace sobre una superficie rugosa con respecto a la longitud de onda. En este caso, el rayo incidente se reflejará en diferentes direcciones en función del ángulo con el que haya incidido sobre la superficie.
2. **Reflexión especular:** este tipo de reflexión se produce cuando el rayo incidente, el reflejado y la normal se encuentran en el mismo plano, siendo la normal perpendicular a la dirección de choque. Se basa en la ley de reflexión total, introducida por W. Snel van Royen en el siglo XV y establece que el ángulo del rayo reflejado es igual al incidente con respecto a la normal a la superficie que la originó.

En la Figura 2.6 se muestran estos tipos de reflexión en función de la frecuencia o longitud de onda. Hay que tener en cuenta que el tamaño del obstáculo o superficie así como la longitud de onda, determinan si existirá una reflexión o la onda será capaz de bordear el obstáculo. Si el obstáculo es pequeño con respecto a la longitud de onda, existirá difracción (véase sección 2.3.3), si por el contrario, el obstáculo es grande con respecto a la longitud de onda existirá reflexión.

Cuando se produce una reflexión especular se cumple que tanto el rayo incidente como el reflejado y la normal al plano de incidencia se encuentran en el mismo plano. Además, el ángulo que forma la onda incidente con la normal es igual al ángulo de la onda reflejada. En el caso que una onda incida en una superficie de forma perpendicular a ella se reflejará sobre sí misma.

Existen diferencias de comportamiento de las ondas al incidir sobre un obstáculo o superficie, como se comprueba en la Figura 2.6. Si la onda sonora tiene alta frecuencia, es decir, baja longitud de onda,



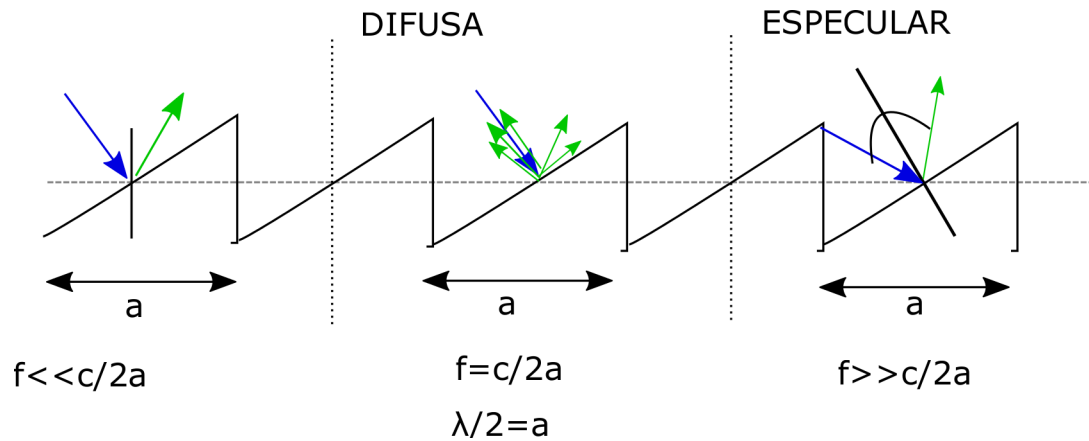


Figura 2.6: Tipo de reflexión en función de la frecuencia. (Fuente: basada en [2])

no rodeará el obstáculo, sino que se reflejará y puede producir sombras en recintos que hay que tener en cuenta. Si por el contrario, la onda tiene baja frecuencia rodeará fácilmente los obstáculos existentes en el recinto.

### 2.3.3 Difracción

La difracción [14] es el fenómeno que sucede cuando el rayo sonoro se dispersa al encontrarse con un obstáculo o al atravesar una abertura. Se basa en el Principio de Huygens que establece que cualquier punto de un frente de ondas puede convertirse en nuevo foco emisor de ondas idénticas a la que lo originó, denominadas ondas difractadas. En la Figura 2.7 se puede observar un obstáculo con el que se encuentra la onda incidente, observando una onda difractada, aunque existirán varias.

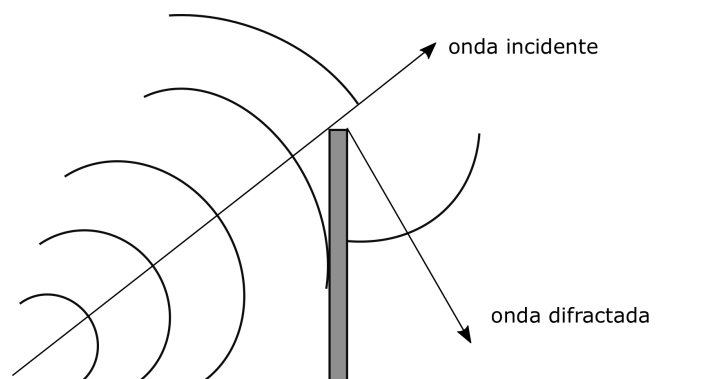


Figura 2.7: Fenómeno de difracción en un obstáculo

Al igual que se ha especificado en el fenómeno de reflexión, la aparición del fenómeno de difracción depende en gran medida de la relación entre el tamaño de los obstáculos y la longitud de onda con que emita la fuente. Si se dispone de un obstáculo de gran tamaño en comparación con la longitud de onda, el efecto de la difracción es muy pequeño y la onda se propaga en líneas rectas o rayos. Si el tamaño del obstáculo es comparable a la longitud de onda, los efectos de la difracción son grandes y se produce la dispersión de los rayos.

En el caso del sonido, las longitudes de onda del espectro audible se sitúan entre 1,7 cm y 17 m y, por tanto, las frecuencias sonoras bajas son más capaces de bordear obstáculos que las altas. Visto desde el punto de vista contrario, si existe un obstáculo de gran tamaño, el rayo no podrá bordear el obstáculo y no se dará el fenómeno de la difracción. Las longitudes de onda del sonido audible son grandes comparadas

con los obstáculos (por ejemplo puertas o ventanas), por tanto, la desviación de las ondas rodeando las esquinas es un fenómeno común.

Especial atención merece en este trabajo la difracción producida en los bordes de los obstáculos, conocida en inglés como *edge diffraction*. En ésta los bordes dispersan parte de la energía sonora en todas direcciones. Ante el análisis acústico de un recinto cerrado, los obstáculos más comunes suelen ser ventanas, mesas, etc. Estos obstáculos tienen una dimensión mayor que la longitud de onda del sonido, por lo que la difracción es un fenómeno existente con frecuencia en este tipo de recintos. La mayoría de algoritmos de cálculo de respuestas al impulso existentes no tienen en cuenta la difracción en bordes ya que asumen la propagación del sonido como la de la luz. Existen multitud de métodos y artículos con diferentes enfoques para calcular las respuestas al impulso de las trayectorias con difracción en bordes en un recinto, a continuación se destacan algunos de ellos:

1. Aproximación de Kirchoff [15]: Huygens fue el primero en considerar el fenómeno de difracción pero no llegó a realizar una formulación matemática del mismo. Fue en 1882 Gustav Kirchoff quien expresó este fenómeno de forma matemática con una fórmula conocida como fórmula de difracción Fresnel-Kirchoff, la cual modela la propagación de los rayos luminosos obteniendo una solución numérica. Para resolver esta ecuación es necesario realizar una serie de aproximaciones y por tanto, las difracciones calculadas tienen algunos errores. A menores frecuencias los errores son más notables, no obstante para determinados ángulos de incidencia en frecuencias mayores también arroja resultados erróneos.
2. Método de Biot-Tolstoy: este método fue el primero en obtener resultados exactos de difracción suponiendo bordes infinitos.
3. Método *Biot-Tolstoy-Medwin* (BTM): descrito por Medwin et al., introdujeron al método de Biot-Tolstoy bordes finitos dando a conocer el método resultante por las iniciales de los autores. En un recinto cerrado los bordes son finitos, por ello es importante realizar una solución de esta forma. No hace suposiciones ante altas frecuencias en las que las longitudes de los bordes y las distancias de la fuente y receptor son mucho mayores que la longitud de onda, por lo que puede arrojar valores poco exactos.
4. Método Svensson et al. [16–18]: desarrolla el modelo BTM llegando a justificar las suposiciones de Medwin para modelar la difracción de primer orden y mejora el cálculo en difracciones de segundo orden. Este método modela la difracción en el dominio temporal junto con el método fuente imagen y arroja resultados casi idénticos en geometrías simples a las medidas de un modelo a escala. Se puede aplicar a bordes curvados y rígidos, así como a difracción con orden múltiple. Ante bordes no totalmente rígidos el método no es exacto.
5. Vanderkooy [19]: junto a Svensson añaden variaciones para mejorar la exactitud de los resultados.

### 2.3.4 Refracción

La refracción [20] es el fenómeno que produce una desviación de las ondas sonoras en la dirección de su propagación cuando pasan de un medio a otro con distintas propiedades. Suele suceder cuando el medio no es homogéneo, es decir, las propiedades varían de una posición a otra y por tanto, la velocidad de propagación de las ondas sonoras varía según se detalla en la ecuación 2.2.

Dentro del campo de la acústica de recintos, el medio no homogéneo aparece en múltiples ocasiones debido a la existencia de sistemas de acondicionamiento de temperatura entre la fuente y el oyente. Estos

sistemas crean columnas de aire a distinta temperatura que la ambiente provocando la aparición del fenómeno de refracción como se puede comprobar de forma esquemática en la Figura 2.8. Este fenómeno no es contemplado en el sistema de simulación que se desarrolla en el presente trabajo ya que para simular este tipo de fenómeno es necesario un sistema de tiempo real capaz de conocer las variaciones instantáneas de temperatura en los distintos puntos del recinto.

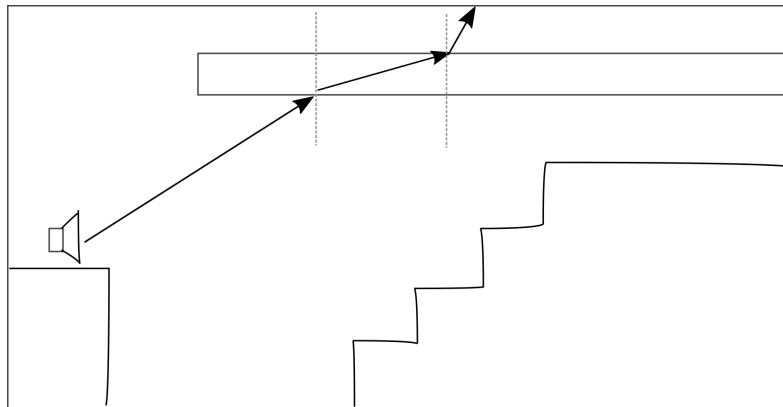


Figura 2.8: Fenómeno de refracción en un recinto con sistema de acondicionamiento de temperatura

## 2.4 Teorías acústicas

Como se ha comprobado a lo largo del capítulo, las ondas sonoras al tomar contacto con las superficies y obstáculos de un recinto sufren modificaciones en la dirección, espectro y magnitud de onda. Estas modificaciones dependen de factores como el ángulo de incidencia, la relación entre la longitud de onda y el tamaño del obstáculo o las características del medio, entre otras. En el campo de la acústica es interesante modelar el campo sonoro en un recinto cerrado por diversos motivos, y es precisamente esto en lo que se basa el sistema de simulación desarrollado.

Para modelar el campo acústico de un recinto se comenzaron usando modelos a escala que requerían un coste importante. En los años 60 comenzó a adquirir relevancia el modelado computacional de la acústica de recintos gracias a Schroeder [21], llevado a la práctica aproximadamente una década después por Krokstad et al. [22] y el mismo Schroeder.

Para hallar el campo acústico de un recinto es necesario resolver las ecuaciones diferenciales de la onda sonora en el espacio y tiempo (ecuaciones de Helmholtz), que como es sabido, resulta una tarea complicada obtener soluciones exactas de las mismas. Por ello, se realizan soluciones aproximadas mediante simulaciones.

En la actualidad, existen tres enfoques diferentes para modelar la acústica de un recinto en función del grado de precisión y complejidad de la determinación del campo sonoro [23] y que se enumeran a continuación:

1. Métodos basados en la teoría estadística
2. Métodos basados en la teoría geométrica
3. Métodos basados en la teoría ondulatoria

### 2.4.1 Teoría estadística

Los métodos que forman parte de la teoría estadística son capaces de estudiar el comportamiento de la energía acústica en un recinto de forma rápida y sencilla, permitiendo obtener medidas cualitativas que permiten detectar fallos en la acústica del mismo. Para que las teorías estadísticas tengan validez, el problema debe cumplir una serie de criterios:

1. Disponer de un campo acústico uniforme en el recinto, de modo que la llegada de ondas a cada punto del mismo sea equiprobable, consiguiendo así un campo sonoro difuso.
2. Las dimensiones del recinto deben ser superiores a la longitud de onda.
3. Los coeficientes de absorción de las superficies del recinto deben ser similares.
4. Los resultados deben ser valores promedio y no instantáneos.

El único método que se resaltarán dentro de la teoría estadística es el método de análisis de energía estadístico conocido como *Statistical Energy Analysis* (SEA), que se describe a continuación:

El método de análisis de energía estadístico [24, 25], conocido por sus siglas en inglés como SEA, predice comportamientos de sonido en sistemas acústicos complejos. Es de utilidad para simulaciones acústicas en etapas iniciales de un diseño ya que el tiempo en realizarlas es menor con respecto a otros métodos. Además, se aplica en la predicción de respuestas en altas frecuencias por su buen funcionamiento en estos casos.

En este método, el sistema se particiona en subsistemas acoplados y se define un conjunto de ecuaciones diferenciales lineales que describen la entrada, almacenamiento, transmisión y disipación de energía dentro de cada subsistema. Los parámetros en las ecuaciones se obtienen haciendo suposiciones estadísticas sobre propiedades dinámicas locales de cada subsistema, simplificando considerablemente el análisis y haciendo posible analizar la respuesta de sistemas que son demasiado complejos para analizarlos por otro método no estadístico.

El método SEA consiste a grandes rasgos en las siguientes etapas:

1. Particionar el sistema en subsistemas más pequeños (como superficies, materiales o cavidades acústicas) acoplándolos entre sí mediante «uniones».
2. Definir el campo reverberante como un grado de libertad en las ecuaciones SEA. La capacidad de almacenar energía en cada campo reverberante se denomina «densidad modal» y depende de la velocidad media con la que se propagan las ondas de energía en el subsistema y de la dimensión global del mismo.
3. Definir la transmisión de energía entre los diferentes subsistemas a través de las «uniones» con el parámetro denominado «factor de pérdidas de acoplamiento», obteniendo los campos directos y reverberantes. Para simplificar los cálculos de este factor, a menudo se asume que hay dispersión dentro de cada subsistema, por lo que las transmisiones entre campos directos es insignificante y dominan los campos reverberantes.
4. Obtener los niveles de presión sonora y las velocidades de vibración medias a partir de los campos definidos.

En términos prácticos, este método es adecuado de aplicar en problemas en los que se dispone de un sistema de gran tamaño en comparación con la longitud de onda sonora.

### 2.4.2 Teoría geométrica

Los métodos basados en la teoría geométrica están basados en rayos. Utilizan el concepto de rayo sonoro definido como una porción de una onda esférica con un ángulo de apertura infinitesimal que tiene como origen el centro de la esfera [10]. El rayo va a determinar la trayectoria de la onda desde que es emitida en la fuente hasta que llega al oyente. La trayectoria dependerá de los diferentes fenómenos vistos en el apartado 2.3 y del medio de propagación.

Las técnicas geométricas calculan la respuesta impulsiva del recinto determinando reflexiones de rayos especulares, y en algunos casos reflexiones difusas y difracciones en bordes. La mayoría de estos métodos se basan en la cantidad de energía, a excepción de los métodos que se ocupan solamente de reflexiones especulares y a difracciones de borde que también pueden ser calculados por métodos basados en presión.

Los dos métodos más populares de la teoría geométrica es el fuente imagen (*image source*) y el método de trazado de rayos (*ray tracing*). Existen otros métodos como el de trazado de haz (*beam tracing*) y el trazado de conos (*cone tracing*), ambos derivaciones de los anteriores. Otro método menos popular que los anteriores, es el conocido como *frustum tracing* [26] que se trata de una aproximación de los anteriores, reduciendo el coste computacional en geometrías complejas y con fuentes y receptores en movimiento.

#### 2.4.2.1 Método fuente-imagen

El método fuente-imagen [23, 27, 28], conocido por sus siglas en inglés como *Image-Source Method* (ISM), es capaz de determinar con precisión la trayectoria y dirección que han seguido los rayos sonoros desde que fueron emitidos en el origen hasta que son recibidos por el oyente. Es aplicado principalmente cuando se desea un modelo con reflexiones tempranas especulares, ya que a medida que aumenta el número de reflexiones a calcular, aumenta exponencialmente el coste computacional.

- **Origen:** este método fue desarrollado para su aplicación al estudio de la electro-estática, y rápidamente fue aplicado al campo de la acústica por Carslaw en 1899. Las primeras aplicaciones en acústica del método abordaron problemas de superficies infinitas y su extensión a recintos con forma de paralelepípedo, presentada por Cremer en 1948. Más tarde, en 1950, Mintzer lo aplicó al cálculo de la presión del sonido en un recinto rectangular en función del tiempo. Gibbs y Jones fueron los primeros en utilizar ordenadores para determinar la posición de las fuentes imagen a través de este método.
- **Funcionamiento:** el método fuente imagen destaca por su sencillez y la obtención de soluciones exactas para recintos rectangulares con superficies rígidas. En este tipo de recintos, idealmente deberían existir únicamente reflexiones especulares y no existiría difracción debido a que las esquinas tendrían ángulos rectos. No obstante, con la existencia de obstáculos en el recinto aparece el fenómeno de difracción de bordes y debe ser calculado.

Se trata de un método recursivo, por tanto, es fácilmente automatizable siguiendo estos pasos:

1. Suponiendo un recinto rectangular, con una fuente de sonido en su interior, el primer paso consiste en reflejar la fuente en todas las superficies del recinto, obteniendo así un conjunto de fuentes imagen (conocidas como *Image Source*, IS).
2. El siguiente paso, e iterativo, consiste en seleccionar cada una de las fuentes imagen obtenidas y volver a reflejar cada una de ellas con las superficies que la rodean.
3. Debe existir una condición de terminación para que el método concluya, bien sea por el orden de reflexión deseado o bien por la longitud máxima que recorran los rayos.

El funcionamiento del método fuente imagen puede ser explicado desde otro punto de vista. Se puede considerar que los rayos reflejados en una superficie se pueden reemplazar por rayos directos que provienen del conjunto de fuentes imagen  $S'$  o fuentes virtuales (*Virtual Source* (VS)). En la Figura 2.9 se muestra un esquema del funcionamiento de este sencillo método. Se puede observar un recinto rectangular en el que se sitúa una fuente ( $S$ ) y un receptor ( $R$ ), teniendo en cuenta que la superficie derecha de este recinto es rígida. La posición de la fuente imagen ( $S'$ ) se calcula teniendo en cuenta que tiene que estar a la misma distancia de la superficie reflejante que la fuente original, es decir,  $S'$  debe posicionarse equidistante de la fuente real ( $S$ ) según la normal a la superficie que le da origen. De esta forma, la fuente imagen se convierte en fuente de un rayo directo hasta el receptor como se había adelantado.

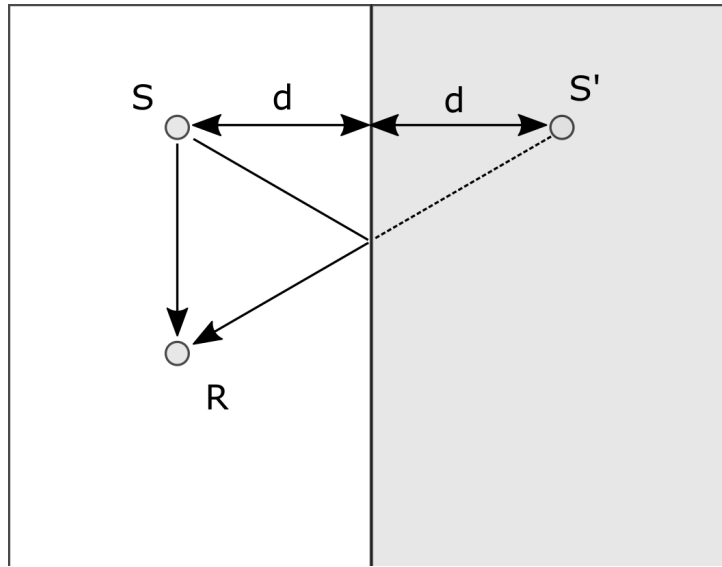


Figura 2.9: Esquema del método fuente-imagen

Con este método, el campo sonoro del recinto está formado por el rayo directo y el conjunto de rayos originados en las fuentes imagen. En la Figura 2.10 se muestra el rayo directo en color rojo, así como 4 rayos reflejados de orden 1 que llegan al oyente tras haberse reflejado en una superficie. Se asume que las superficies interiores del recinto son rígidas (reflejan toda la energía que incide en ellas) y que la fuente y el receptor son omnidireccionales.

- **Algoritmo matemático:** se ha especificado que la fuente imagen se encuentra equidistante de la fuente real con respecto a la normal que las separa, no obstante, desde un punto de vista matemático el cálculo de la posición de la fuente imagen se realiza del siguiente modo:

1. Determinar la ubicación del plano reflejante en el espacio: su ubicación se define por la normal unitaria al plano  $\hat{n}$  y, su distancia  $p$  al origen de coordenadas del recinto. Así, en la ecuación 2.10 se calculan las coordenadas de la posición en el espacio de la fuente imagen:

$$s_v = s_r + 2d\hat{n} \quad (2.10)$$

siendo

$$d = p - s_r \cdot \hat{n} \quad (2.11)$$

donde  $d$  es la distancia de la fuente imagen a la superficie y  $s_r$  representa las coordenadas de la posición en el espacio de la fuente real.

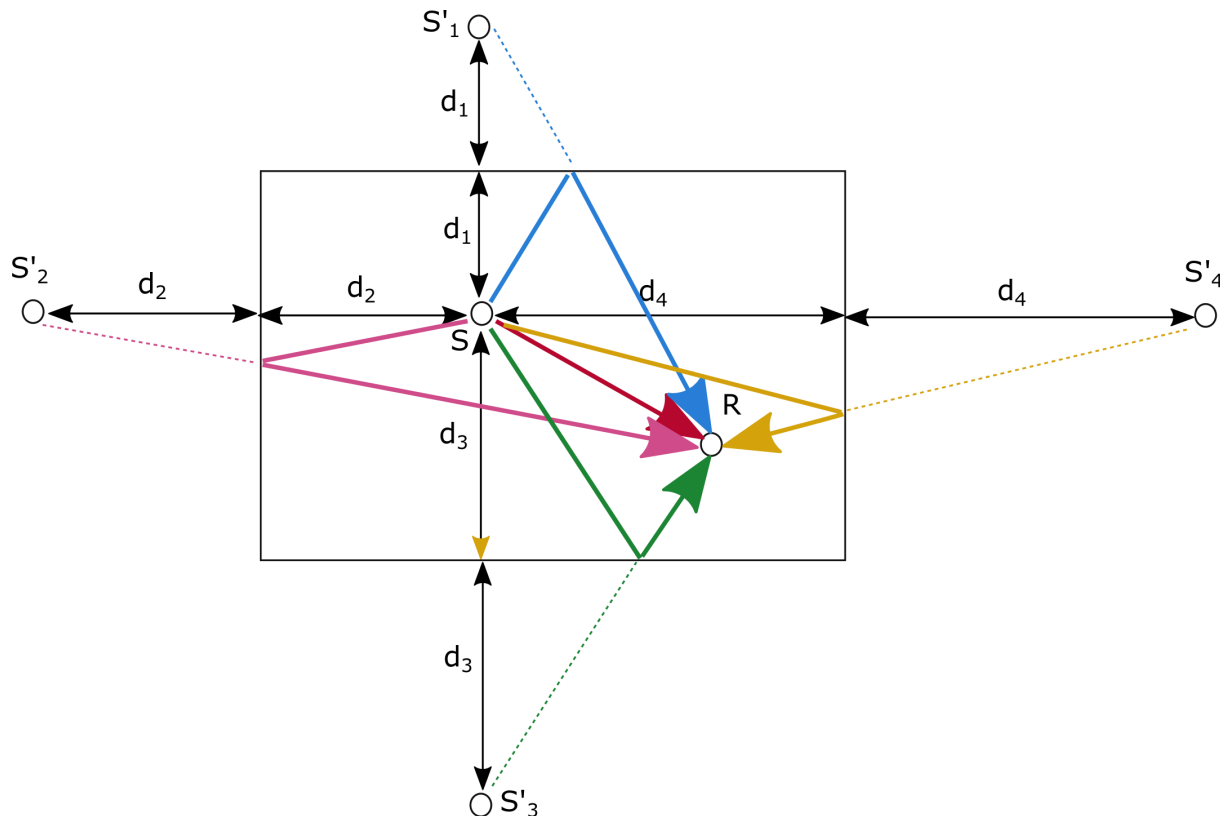


Figura 2.10: Aplicación método fuente-imagen

En la Figura 2.11 se pueden comprobar cada una de las variables que intervienen en el cálculo matemático de la posición de la fuente imagen.

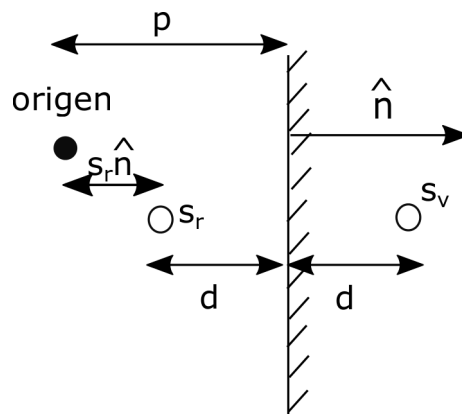


Figura 2.11: Posición de VS. (Fuente: basado en [1])

- Una vez calculadas las posiciones de las fuentes imagen reflejando la fuente real a través de cada superficie reflejante del recinto, según el orden de reflexión elegido, cada una de estas fuentes imagen se vuelve a reflejar en cada una de las superficies que la rodea para generar los ordenes de reflexión deseados. Por este motivo, desde el punto de vista matemático del algoritmo, se genera una estructura tipo árbol que organice cada uno de los órdenes. El primer reflejo de la fuente real genera las fuentes imagen primarias, cada una de estas, generará fuentes imagen secundarias, y así sucesivamente. Si nos encontramos ante un recinto de  $n$  superficies, habrá  $n$  posibles fuentes imagen de primer orden. Cada una de estas fuentes imagen primarias dará

origen a  $(n - 1)$  fuentes imagen de orden superior. Es por este motivo que según se aumenta el orden de reflexión, aumenta de forma exponencial  $n(n - 1)^{i-1}$  el coste computacional del algoritmo.

- **Comprobaciones:** el procedimiento del método fuente imagen como se ha comprobado es sencillo de aplicar en recintos con forma de cuboide, pero para formas arbitrarias existe el problema de que algunas fuentes imagen de orden superior pueden no contribuir al campo sonoro total, por este motivo, se deben realizar las siguientes pruebas:

1. **Validez:** comprobación de que la fuente imagen fue creada por una superficie reflejante, en caso contrario, no será válida. Desde el punto de vista matemático, una fuente imagen no es válida si la distancia obtenida en la ecuación 2.11 es negativa. Esto indicaría que la fuente imagen calculada está dentro del recinto estudiado.
2. **Proximidad:** prueba necesaria para finalizar el algoritmo recursivo de este método ya que es necesario terminar la generación de nuevas fuentes imagen. Una forma de finalizar es especificar la longitud máxima que pueden recorrer los rayos y en caso de superarla se descartan, y otra forma es especificar el orden máximo de las fuentes imagen. En el caso del sistema de simulación desarrollado, se ha optado por la segunda de las opciones, es decir, antes de comenzar se selecciona el orden de reflexión deseado y así se conoce a priori el número de fuentes imagen a calcular.
3. **Visibilidad:** trata de averiguar si la fuente imagen creada es visible para el oyente a través de la superficie que la creó. Para fuentes imagen de primer orden, la fuente imagen es visible si la recta trazada entre ésta y el oyente pasa por el plano de la superficie reflejante, en un punto que está dentro de los límites de la superficie. En la Figura 2.12 se muestra el caso en el que el punto de intersección entre la fuente imagen y el receptor está fuera de la superficie y debería ser descartada. Desde otro punto de vista, la visibilidad comprueba que el camino entre la fuente imagen y el receptor debe cruzar todas las superficies implicadas en la generación de la fuente imagen y no otras, y la última pared que atraviesa debe ser una pared física y no virtual.

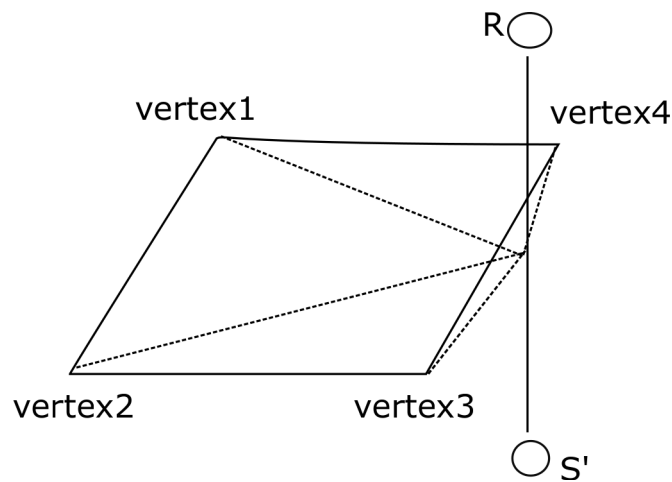


Figura 2.12: Comprobación de visibilidad entre VS y oyente. (Fuente: basado en [1])

En el caso que las fuentes imagen calculadas tengan más de un orden se requieren realizar otras pruebas de visibilidad adicionales. En la referencia [28] se indica que los trayectos que involucran más de una reflexión requieren el cálculo de oyentes virtuales de orden superior que



apuntan hacia su fuente imagen. La desventaja de esta prueba es que requiere de un gran coste computacional.

Cuando una fuente imagen incumple las pruebas de validez o proximidad es descartada, sin embargo, si no cumple la prueba de visibilidad puede que no haya que descartarla ya que las fuentes imagen de mayor orden pueden sí ser visibles.

#### 2.4.2.2 Método de trazado de rayos

El método del trazado de rayos, conocido en inglés como *ray-tracing* [22,23], es una derivación del método fuente imagen basado en superar el principal problema de éste. El método fuente imagen busca todos los rayos existentes con reflexión especular y como se ha visto, se establecen unas condiciones de terminación para acabar con el algoritmo. En su lugar, el método del trazado de rayos no busca todos los trayectos de forma determinista, sino que utiliza métodos estocásticos y un tipo de simulación *Monte Carlo* en todos los posibles rayos reflejados, de este modo, limita la computación asumiendo que la fuente emite un número finito de rayos uniformemente distribuido por todas direcciones, obteniendo así un número finito de rayos.

Este método además de encontrar las reflexiones especulares, permite determinar las reflexiones difusas. El primero en describir este aspecto fue Kuttruff en [10].

- **Origen:** la técnica de trazado de rayos se remonta a 1958 cuando Allred y Newhouse publicaron un estudio sobre el uso de esta técnica para detectar rayos en un espacio cerrado rectangular. Sin embargo, la aplicación del método a la obtención de respuestas al impulso en función del tiempo en recintos en 3D fue descrito por Krokstad et al. en [22] en 1968. En paralelo, Schroeder presentó ideas similares pero su técnica estaba más limitada y operaba solo en espacios en 2D.
- **Funcionamiento:** el principio básico de este método es lanzar rayos desde una fuente sonora, reflejarlos en las superficies del recinto, y registrar los rayos válidos. Los rayos pueden ser emitidos desde la fuente de dos formas diferentes:
  - de acuerdo a una distribución predefinida o,
  - con direcciones aleatorias a través del método *Monte Carlo*.

El uso de direcciones aleatorias con muy pocos rayos introduce variaciones en los resultados y por tanto, es más conveniente emplear el primero de los métodos, con una distribución predefinida.

La propagación de cada rayo se determina estableciendo que cada vez que un rayo golpea una superficie, se produce una reflexión y el rayo continua en una nueva dirección. Al mismo tiempo, se calculan las intersecciones de los rayos con los oyentes para determinar que rayos reflejados llegan al receptor desde la fuente sonora.

En este método debe definirse el momento en el que los rayos no continúan su trayectoria y se detienen, al igual que en el método de fuente imagen. Este momento se encuentra relacionado con la atenuación de la energía de la onda. Se resaltan dos métodos para definir este momento:

1. Cada rayo transporta información sobre su energía, y siempre que es reflejado, la energía se atenúa de acuerdo a las propiedades del material reflector. Esta atenuación depende también de la frecuencia de la onda sonora, sin embargo, la variación de atenuación puede dar lugar a que las trayectorias de los rayos no converjan. La trayectoria del rayo termina cuando la energía decae por debajo de un valor dado para todas las bandas de frecuencia, o cuando se supera una distancia máxima de propagación.

2. Este otro método se basa en la aplicación de la «ruleta rusa». En cada reflexión, un rayo tiene una cierta probabilidad de no continuar su trayectoria basándose en los coeficientes de absorción del material. Este enfoque no es adecuado para trazar rayos simultáneamente en diferentes bandas de frecuencia y además, no es tan eficiente como el primero.

Para determinar con el método de trazado de rayos las reflexiones difusas se procede del siguiente modo:

1. Se divide la energía reflejada en dos partes: una procedente de reflexión especular y otra de reflexión difusa. Se asigna a cada una de estas divisiones una tasa que depende del coeficiente de dispersión.
2. A la hora de trazar los rayos difusos se puede realizar de dos formas diferentes:
  - Los rayos se pueden dividir en cada reflexión en dos componentes, un rayo especular primario y numerosos rayos reflejados de forma difusa igualmente distribuidos en el medio en dirección opuesta a la superficie.
  - Se puede mantener un único rayo y en cada reflexión, el rayo viaja en una dirección aleatoria si el coeficiente de dispersión de la superficie reflejante es mayor que un número determinado al azar.

La primera opción es una buena aproximación pero tiene un elevado coste computacional ya que el número de rayos se incrementa de forma exponencial. La segunda opción requiere menor coste computacional pero también supone la obtención de resultados menos precisos.

- **Aspectos de relevancia:** uno de los problemas principales del método de trazado de rayos viene motivado por la probabilidad que tiene un rayo de pasar a través del receptor y ser considerado como contribuyente a la respuesta total. En este método, se realiza un recuento de los rayos que atraviesan al receptor, considerando a éste último como un detector volumétrico. Es precisamente el volumen que se le asigne una cuestión de relevancia ya que si se considera al receptor como un detector volumétrico puntual, la probabilidad de que un rayo atravesase ese punto es infinitamente pequeña. Por eso, se le asigna al detector un volumen mayor pero esto puede traer problemas en cuanto a que se pueden detectar trayectos no válidos y registrar algunos rayos reflejados múltiples veces.

Una de las posibles soluciones es disponer de detectores volumétricos con tamaño variable, incrementando el volumen linealmente en función de la distancia de propagación. Pero esta solución podría constituir otro método denominado como trazado de haz y explicado en el apartado 2.4.2.3.

Otra solución a menudo ventajosa en la práctica explicada en [22], es disponer de varios receptores en la simulación. De esta forma se registrarían de forma acumulativa cada uno de los instantes en los que un rayo pasa a través de un receptor. Permite considerar al receptor como una esfera omnidireccional sin usar ningún tipo de ponderación pero tendría que ser ponderados los ángulos de incidencia de las reflexiones en las superficies con los receptores existentes.

También es posible obtener más precisión en los resultados teniendo en cuenta la longitud de la trayectoria de los rayos viajan en el interior del receptor, pudiendo usar receptores de cualquier forma y no solo esféricos.

Es importante destacar también que la precisión de la solución por trazado de rayos aumenta al incrementar el número de rayos calculado, de tal manera que los resultados de la solución del método de fuente imagen y trazado de rayos convergen en un escenario en el que solo se consideran reflexiones especulares. La mayor dificultad de esta técnica es saber el número de rayos necesario para resultados fiables ya que depende de la geometría o los materiales, entre otros aspectos.

### 2.4.2.3 Método de trazado de haz

El método de trazado de haz, conocido en inglés como *beam tracing* [29,30], deriva del método de trazado de rayos, con una diferencia fundamental, y es que sustituye a un rayo de dicho método por uno o varios formando un haz. La diferencia entre un rayo y un haz es que el primero no tiene espesor. Un haz tienen forma de pirámide no acotada con secciones transversales poligonales.

Este método se basa en la emisión de un haz piramidal a través del tronco del cono de visión. Al cruzarse con un obstáculo o superficie absorbente del recinto, se elimina parte del haz y se añade a una cola de procesamiento. Si la superficie u obstáculo es de un material reflejante, se crea un nuevo haz.

### 2.4.2.4 Método de trazado de conos

El método de trazado de conos, conocido en inglés como *cone tracing* [31], deriva del método de trazado de rayos al igual que el método de trazado de haz y al igual que éste consiste en que emite rayos con espesor. Surge como una solución al problema existente en el método de trazado de rayos del solapamiento (*aliasing*) por estar ante señales geométricas proyectadas. Si las frecuencias son muy altas, superan a la frecuencia máxima de Niquist y surge el solapamiento.

### 2.4.2.5 Comparativa

Como se ha adelantado, los métodos más populares son el método fuente imagen y el método de trazado de rayos, siendo el resto de métodos mejoras de este último pero con la misma base teórica. Además, como se ha explicado, el método de trazado de rayos puede considerarse un caso particular de fuente imagen. Por tanto, son múltiples las similitudes entre los métodos dentro de la teoría geométrica.

Centrando la atención en la longitud de onda, el método fuente imagen se suele aplicar en casos en los que la longitud de onda es pequeña con respecto al tamaño de las superficies del recinto, y grande en relación a las irregularidades y curvaturas de las mismas.

En cuanto al cálculo de trayectorias, el método fuente imagen busca todas las fuentes imagen existentes, dibuja trayectorias rectas entre fuente receptor y comprueba la visibilidad. En el método de trazado de rayos, el rayo sale de la fuente sonora y viaja por la sala, cambia de dirección y pierde energía en cada reflexión, llegando al receptor con una determinada energía y en un instante de tiempo concreto. En el método fuente imagen el receptor es considerado un punto mientras que en trazado de rayos se considera como un volumen, aspecto considerado como un inconveniente, pero como se analizará a continuación, superado por el resto de métodos.

Cabe destacar que el método de trazado de rayos es más sencillo de programar que el método fuente imagen, sobretudo en recintos con geometrías complejas. Además, es posible determinar reflexiones difusas mientras que el método fuente imagen sólo es capaz de determinar reflexiones especulares. En [32,33], el autor realiza un estudio de comparación de ambos métodos en los que demuestra que el método de trazado de rayos tiene mejor rendimiento en casi todas las situaciones que fuente imagen.

Por su parte, el mayor inconveniente del método de trazado de rayos es la determinación correcta del volumen del receptor para detectar los rayos que recibe. Como se ha visto, los métodos de trazado de haz y trazado de conos, tratan de superar este inconveniente. Además, el método de *frustum tracing*, como se adelantó, combina todos los métodos anteriores para reducir el coste computacional de recintos con geometrías arbitrarias y ante la existencia de fuentes y/o receptores en movimiento.

### 2.4.3 Teoría ondulatoria

Los métodos basados en la teoría ondulatoria permiten estudiar la naturaleza ondulatoria del sonido, resolviendo su ecuación de onda formulada en la ecuación 2.3. Para realizar este estudio, tratan de resolver un conjunto de ecuaciones diferenciales parciales de gran complejidad que en la mayor parte de las ocasiones se trata de aproximaciones numéricas.

Estos métodos consideran el volumen de aire del interior de un recinto como un sistema vibratorio complejo, constituido por un conjunto de subsistemas de menor complejidad. La fuente sonora genera vibraciones forzadas y modos de vibración, que, al alcanzar la situación estacionaria o de equilibrio se habrán amortiguado. En equilibrio, las vibraciones se expresan como un gran número de ondas estacionarias, mientras que cuando la fuente sonora deja de emitir las vibraciones y modos de vibración se amortiguan gradualmente de forma exponencial. Por tanto, los métodos de esta teoría permiten estudiar la influencia de las dimensiones y proporciones del recinto en la acústica del mismo.

#### 2.4.3.1 Método de elementos finitos

El método de elementos finitos [34, 35], conocido por sus siglas en inglés como *Finite Element Method* (FEM), trata de encontrar una solución numérica aproximada a las ecuaciones diferenciales parciales que definen una estructura. Obtiene resultados adecuados aun en geometrías complejas.

El nombre de este método es debido a que divide (discretiza) el dominio de estudio en multitud de subdominios no intersectantes entre sí denominados «elementos finitos». Cada uno de estos subdominios está formado por un conjunto de nodos y los nodos adyacentes forman las conocidas como «mallas». Se considera que dos nodos son adyacentes si pertenecen al mismo elemento finito. Los cálculos se realizan sobre las «mallas», siendo considerada cada una de ellas un elemento finito.

Este método al realizar particiones tiende a la convergencia ya que si las particiones cada vez son más finas la solución convergirá a una solución más exacta. A modo de resumen de este método, se detallan a continuación las etapas requeridas para resolver un problema por el método FEM:

1. El problema a resolver debe estar caracterizado por ecuaciones constitutivas y ecuaciones de evolución temporal formuladas de forma débil, es decir, descritas en forma integral que permitan ser resueltas por métodos de álgebra lineal. En otras palabras, el espacio a simular debe estar definido por unas ecuaciones diferenciales parciales.
2. El dominio espacial o temporal definido tiene que ser dividido en subdominios llamados «elementos finitos». Así, se construye un espacio vectorial de dimensión finita, denominado espacio de elementos finitos. La solución aproximada se obtiene a través de estos elementos finitos.
3. El tercer paso consiste en proyectar el problema original sobre el espacio de elementos finitos que se ha obtenido en la discretización del espacio. De este modo se obtiene un número finito de ecuaciones, con tantas incógnitas como dimensiones del espacio vectorial de elementos finitos existan.
4. En el último paso se realiza el cálculo numérico de la solución del sistema de ecuaciones.

La clave de este método se encuentra en la discretización del dominio, ya que esto permite disponer de un algoritmo de proyección sencillo y cuya solución sea exacta en un conjunto finito de puntos del dominio. Estos puntos donde la solución es exacta suele ser en los denominados nodos, mientras que en el resto de puntos del dominio la solución es la interpolación de los resultados exactos obtenidos en los nodos.

En la Figura 2.13 se muestra de forma esquemática los pasos necesarios para resolver un problema por el método FEM.

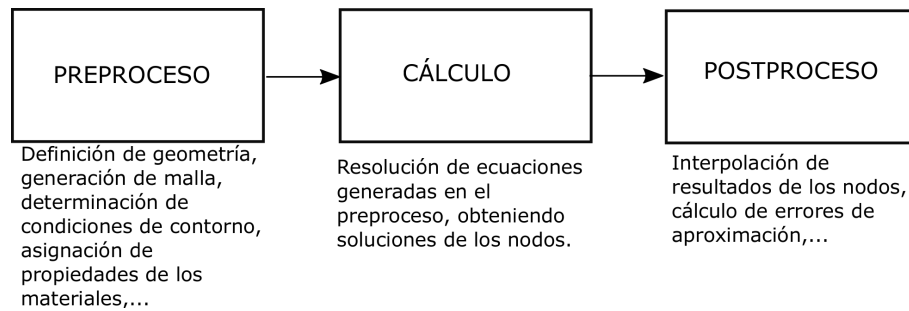


Figura 2.13: Pasos principales del método FEM

### 2.4.3.2 Método de elementos de contorno

El método de elementos de contorno o de elementos de frontera [36], conocido por sus siglas en inglés como *Boundary Element Method* (BEM), trata de resolver ecuaciones en derivadas parciales lineales que han sido formuladas como ecuaciones integrales sobre el contorno de la geometría a estudiar. La característica principal de este método es que utiliza las condiciones de contorno para resolver la ecuación integral en el contorno. De este modo, no calcula todos los puntos del dominio que define la ecuación diferencial parcial. No obstante, en la etapa de postprocesamiento se calcula la solución en los puntos del dominio que no pertenecen al contorno.

Usualmente, para aplicar este método es necesario conocer la función de Green. Esto implica la necesidad de disponer de campos lineales en medios homogéneos, restringiendo así los problemas en los que se puede aplicar este método. Hay que destacar que es posible resolver mediante el método BEM campos no lineales pero para ello es necesario introducir integrales de volumen que discretizan el dominio, eliminando una de las ventajas de este método.

### 2.4.3.3 FDTD

El método de diferencias finitas en el dominio del tiempo [37], más conocido por sus siglas en inglés *Finite-Difference Time-Domain* (FDTD) [38], trata de resolver las ecuaciones de Maxwell definidas en derivadas parciales reemplazándolas por un sistema de ecuaciones en diferencias finitas. Este método discretiza en el espacio y en el tiempo las ecuaciones de Maxwell estableciendo un conjunto de ecuaciones en diferencias finitas. Si se eligen adecuadamente los puntos en que se evalúan las componentes de los campos en las ecuaciones, la solución al sistema de ecuaciones satisface las condiciones de borde.

### 2.4.3.4 Comparativa

Una vez se ha realizado una breve descripción de tres de los métodos de la teoría ondulatoria, haciendo mayor hincapié en el método FEM por ser el más popular, se establecen una serie de diferencias entre ellos en este apartado.

Centrando la atención en la cantidad de recursos computacionales necesarios, se puede establecer que el método BEM es más eficiente que el método FEM cuando nos encontramos con problemas con tasa de superficie/volumen baja. El método BEM resuelve los dominios acústicos solo en el contorno, reduciendo la dimensión del problema de volúmenes a superficies o de superficies a curvas, con lo que el coste computacional disminuye por la discretización y la dimensión de los sistemas de ecuaciones lineales.

Hay que destacar que en algunos casos el método BEM es menos eficiente que FEM, BEM suele aplicarse en dominios acústicos no acotados o infinitos. Ante bajas frecuencias, el método BEM es la

alternativa al método [FEM](#).

Con respecto al método [FDTD](#), cabe destacar que entre sus principales ventajas está que permite modelar geometrías arbitrarias, incluir diferentes tipos de fuentes con facilidad, programación relativamente simple y cálculos precisos. Entre sus inconvenientes encontramos que tiene un alto coste computacional y que la aplicación de las condiciones de contorno es compleja.

#### 2.4.4 Comparativa

Una vez analizados algunos métodos de cada una de las tres teorías utilizadas para modelar recintos acústicos, se realiza en este apartado una comparación entre los distintos aspectos de cada una de ellas.

Teniendo en cuenta que el oído humano es capaz de oír frecuencias desde los 20 *Hz* a los 20 *kHz*, se puede dividir este rango espectral en cuatro zonas (véase Figura 2.14) en las que funcionarán mejor unas teorías u otras [\[39\]](#):

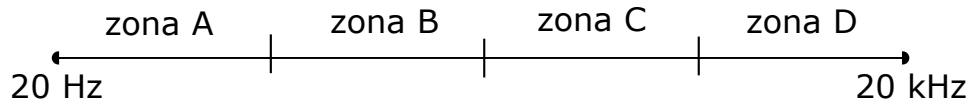


Figura 2.14: División del espectro audible en zonas

- Zona A: no hay modos propios y funciona bien la teoría ondulatoria.
- Zona B: sigue funcionando bien la teoría ondulatoria.
- Zona C: es una zona en la que la frecuencia es demasiado baja para aplicar la teoría geométrica y demasiado alta para aplicar la teoría ondulatoria, por tanto, se puede decir que funciona bien la teoría estadística.
- Zona D: aplicable con resultados aceptables tanto la teoría estadística como la teoría geométrica.

La teoría estadística es de utilidad en etapas iniciales de un diseño, ya que realizando una simulación con la teoría estadística se consiguen tiempos menores de simulación que con otros métodos. Además, se aplica en la predicción de respuestas en altas frecuencias por su buen funcionamiento en estos casos. Los métodos de la teoría estadística se usan mayoritariamente en predicciones de aislamiento entre recintos acoplados. Estos métodos no se suelen aplicar en simulación de recintos porque no son capaces de modelar el comportamiento temporal de las ondas sonoras.

Los métodos basados en teoría geométrica tienen limitaciones en el modelado de ondas sonoras de baja frecuencia, sin embargo, las bajas frecuencias son modeladas adecuadamente por la teoría ondulatoria. Schroeder estableció una frecuencia a partir de la cual se debería de aplicar la teoría geométrica y por debajo de ella la teoría ondulatoria. Esta frecuencia, en la que existe la transición entre las dos teorías, se conoce como frecuencia de Schroeder  $f_s$  y depende del tiempo de reverberación  $T_{60}$  del recinto a modelar y del volumen  $V$  del mismo, según se muestra en la ecuación 2.12.

$$f_s \approx 2000 \sqrt{\frac{T_{60}}{V}} \quad (2.12)$$

Si la frecuencia de la onda sonora es mayor que  $f_s$  existen resonancias con diferentes fases que interfieren entre sí y por ello se prefiere la aplicación de teoría geométrica. Por tanto, la teoría geométrica se prefiere en un campo de alta frecuencia que no es difuso ya que estos métodos ignoran los fenómenos de interferencia

y difracción, ondas estacionarias y modos propios, siendo válida entonces bajo estas condiciones. Como la teoría geométrica sustituye la onda por un rayo que se refleja de forma especular y no envuelve los obstáculos, se aplica en altas frecuencias, es decir, longitudes de onda bajas que sean menores que las dimensiones del recinto.

Las teorías geométrica y estadística estudian el campo sonoro de un recinto mediante idealizaciones, sin embargo, la teoría ondulatoria permite la posibilidad de estudiar la naturaleza ondulatoria del sonido resolviendo la ecuación de onda, aunque con gran complejidad matemática.

Por su parte, los métodos de la teoría ondulatoria son bastante populares pero tienen como principal inconveniente que necesitan una potencia de cálculo muy elevada y su aplicación se restringe entonces a bajas frecuencias.

## 2.5 Parámetros acústicos

Hoy en día existen multitud de sistemas de simulación encargados de determinar los parámetros acústicos de un recinto para obtener una medida de calidad acústica del mismo. La norma ISO 3382 define la medida de parámetros acústicos en recintos en varias partes, una encargada de las salas de espectáculos, otra del tiempo de reverberación en recintos ordinarios y una tercera para oficinas diáfanas.

En el sistema de simulación acústica desarrollado en este trabajo no se ha incluido un estudio de estos parámetros, aunque en el estado del arte de los sistemas de simulación existentes (Capítulo 3) se ha podido comprobar su existencia y por ello se hace un breve repaso de los mismos. Hay que destacar que para la obtención de los distintos parámetros acústicos es necesario obtener previamente la respuesta al impulso del recinto.

### 2.5.1 Coeficiente de absorción sonora

El coeficiente de absorción sonora expresa la cantidad de energía absorbida por un material dependiendo de la frecuencia de la onda emitida. Se define como la relación entre la energía absorbida por una superficie  $E_a$  con respecto a la energía incidente  $E_i$  como se muestra en la ecuación 2.13.

$$\alpha = \frac{E_a}{E_i} \quad (2.13)$$

Este coeficiente además de depender de la frecuencia de la onda, depende del ángulo de incidencia de los rayos reflejados, por lo que el coeficiente de absorción se obtiene promediando los valores de rayos que inciden desde todos los ángulos. Además de conocer la absorción de las superficies de un recinto, en la calidad acústica del mismo influye la absorción de objetos u obstáculos que se encuentren dentro del mismo y que deben ser tenidos en cuenta.

### 2.5.2 Tiempo de reverberación

La reverberación es el fenómeno producido cuando las ondas reflejadas llegan al oyente después de haberse interrumpido la emisión de la onda. El tiempo de reverberación es el tiempo que transcurre desde que deja de emitir sonido la fuente bruscamente hasta que la energía recibida por el receptor está 60 dB por debajo del nivel que recibía. A modo aclaratorio se representa en la Figura 2.15 el tiempo de reverberación  $TR_{60}$ .

En un recinto, cuando una fuente sonora emite un sonido continuo durante un tiempo, la densidad de energía sonora aumenta progresivamente según avanza el tiempo hasta llegar a un estado de equilibrio o régimen permanente en el que la energía absorbida es igual a la emitida. Si estando en este estado estacionario, la fuente deja de emitir sonido, la densidad de energía disminuye gradualmente y no de forma brusca. En la Figura 2.15 se puede observar este fenómeno.

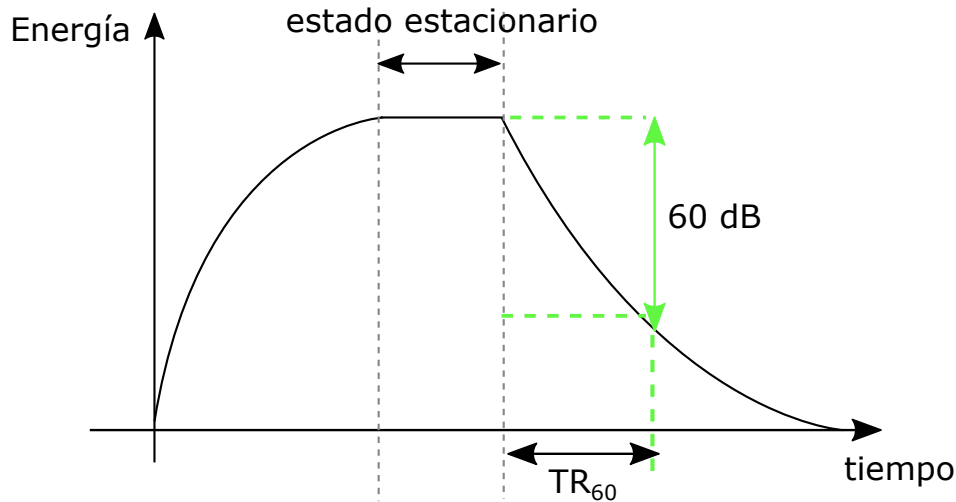


Figura 2.15: Variación de la densidad de energía sonora al emitir un sonido continuo

Existen múltiples definiciones del tiempo de reverberación, una de las más usadas es la fórmula de Sabine, que define el tiempo de reverberación  $TR_{60}$  como:

$$TR_{60} = \frac{0,161V}{Aa} \quad (2.14)$$

siendo  $V$  el volumen del recinto,  $A$  la superficie del recinto y  $a$  la absorción sonora total.

Existen múltiples modificaciones a la fórmula de Sabine, como la introducción de la temperatura, y también nuevas definiciones. El tiempo de reverberación es un parámetro principal para determinar la calidad acústica de un recinto, pero hay que tener en cuenta que su valor depende de múltiples factores como el coeficiente de absorción, la superficie, etc.

### 2.5.3 Claridad

La claridad relaciona la cantidad de energía sonora inmediata que se produce en los primeros 50 ms, para la voz, y 80 ms, para la música, respecto a la energía sonora tardía que se produce después de transcurridos ese tiempo. esta relacionado con la nitidez o claridad de los sonidos en el recinto y cuando mayor sea su valor predominará más el sonido inicial que el final.

## 2.6 Conclusiones

El campo de la acústica ha sido estudiado por multitud de científicos y estos datan desde la Antigüedad. Por tanto, es un área ampliamente investigada en la actualidad para, aprovechando las nuevas tecnologías conseguir resultados cada vez más reales y precisos.

A la hora de realizar un estudio acústico de un recinto se comienza con la elección de un método de los estudiados de las diferentes teorías acústicas. Normalmente, el objetivo es obtener la respuesta impulsiva



del recinto, y a partir de esta calcular los distintos parámetros acústicos que aportan una medida de calidad del mismo.



## Capítulo 3

# Evaluación de sistemas de simulación

### 3.1 Introducción

Uno de los ejes principales en el desarrollo de este trabajo se encuentra en realizar una evaluación de la situación actual respecto a sistemas de simulación acústica. Es por ello que una parte importante de los esfuerzos han sido evaluar, clasificar y seleccionar aquellas herramientas de simulación acústica que cumplieran los requisitos deseados.

Por tanto, este capítulo se divide por un lado en una breve explicación de un conjunto de sistemas de simulación comerciales, realizando una comparación de ellos en base a una serie de características, el mismo procedimiento se lleva a cabo con una serie de sistemas bajo licencia *GNU's Not Unix* (GNU) *General Public License* (GPL). Se centra la atención en estos últimos, realizando la selección de aquel que cumple un mayor número de requisitos exigidos y se describe en mayor profundidad.

### 3.2 Sistemas comerciales

En este apartado de sistemas de simulación acústica comerciales se incluyen todas las herramientas de simulación destacables que poseen licencia de software propietario, según se explica en el Anexo A. Se incluyen por tanto, aquellas herramientas con licencia no gratuita, indicando si existe versión demo del mismo u otro tipo de licencia propietaria. Además, se incluyen en este apartado aquellos paquetes libres que deben ser ejecutados sobre software comercial.

#### 3.2.1 Breve descripción

En este apartado se realiza una breve descripción de una serie de sistemas de simulación de modo que se obtenga una idea general de los principales puntos fuertes de estas herramientas.

1. **ABEC3** (Acoustic Boundary Element Calculator) [40]: herramienta de simulación acústica por medio de métodos de la teoría ondulatoria. Dispone de tres métodos de simulación: método **BEM** para acústicas complejas como recintos, método *Lumped Element Method* (LEM) para acústicas simples como estructuras electroacústicas y un tercer método de cálculo de sonido directo sin tener en cuenta reflexiones ni difracciones. Además de estar orientado a simulación acústica de pequeños recintos, es una buena opción para simulación electromagnética como por ejemplo guías de onda.

2. **AIST-Laboratory** [41]: el paquete AIST Room-Acoustics está basado en el método geométrico de trazado de rayos y es capaz de simular la acústica de recintos con geometría arbitraria. Muestra un diagrama 3D con la trayectoria de los rayos calculados. Existe la posibilidad de descargar una versión libre en la que se restringe a 12 los planos capaces de reflejar la señal. Si se desea eliminar esta restricción ha de ser adquirida una licencia comercial.
3. **Aurora** [42]: conjunto de *plugins* disponibles con licencia libre para el paquete comercial Adobe Audition. Es capaz de simular espacios acústicos en 3D, realizar medidas de parámetros acústicos, auralización en tiempo real u obtener las respuestas al impulso, entre otros aspectos. Ha sido utilizado en teatros y otros espacios.
4. **Cara** (Computer Aided Room Acoustics) [43]: programa para cálculo y optimización de acústica de recintos con geometría arbitraria basado en el método geométrico de fuente imagen. Sus principales funciones son la optimización de la posición de las fuentes sonoras para minimizar interferencias y el cálculo de parámetros acústicos. Es capaz de realizar auralización estéreo en función de la posición de las fuentes.
5. **CATT-Acoustic** [44]: simulador destinado a comprobar el comportamiento acústico de salas. A través de métodos geométricos es capaz de calcular parámetros acústicos y auralización (mono, estéreo, binaural y B-format). La geometría de la sala tridimensional se realiza por medio de un archivo `.cad`.
6. **EASE** (Enhanced Acoustic Simulator for Engineers) [45]: se trata de un programa que ofrece un conjunto de herramientas para simular de forma detallada y realista la acústica de un espacio, así como verificar y evaluar parámetros acústicos. Es capaz de mostrar gráficamente las predicciones exactas, aportar medidas de reverberaciones o la inteligibilidad del hablante, entre otros parámetros. Asegura ser una buena herramienta para analizar distintos parámetros antes de la construcción de un recinto.
7. **LMS RAYNOISE** [46]: sistema de modelado y análisis acústico para entornos abiertos, cerrados o mixtos, todos ellos en 3D. Usa el método de trazado de rayos de la teoría geométrica para predecir el campo sonoro y otros métodos para modelar difusión y difracción. Los resultados pueden ser convolucionados con señales para producir auralización estéreo.
8. **ODEON** [47]: software de acústica de recintos para medición, simulación y auralización. Aunque está principalmente ideado para interiores, puede funcionar en exteriores con resultados aceptables. Usa el método fuente imagen de la teoría geométrica combinado con el método de trazado de rayos.
9. **Olive Tree Lab - Room** [48]: software para modelado y simulación acústica de recintos en 3D que incluye el cálculo de parámetros para evaluar la calidad acústica. Usa el método de trazado de rayos, el usuario puede definir el número de rayos que quiere calcular, y auraliza el sonido de la sala.
10. **RAMSETE** [42, 49]: modela acústica de recintos a través del método geométrico de trazado piramidal, el cual está basado en los métodos de trazado de rayos y trazado de conos, permitiendo resolver problemas de propagación tanto en recintos cerrados como abiertos. Dispone de una herramienta para auralización.
11. **RoomOptimizer** [50, 51]: herramienta para la colocación óptima de fuentes de sonido y receptores en un recinto con geometría cuboide. Combina técnicas geométricas con técnicas de optimización multidimensional.

12. **RPlusD** [52]: es el software más reciente de etfacoustic que sustituye al software ETF5.0 de la compañía. Está destinado a mediciones del sistema de audio de salas de pequeño y mediano tamaño obteniendo una serie de parámetros acústicos. Utiliza el método *Maximum Length Sequence* (MLS) para el cálculo de la respuesta al impulso.
13. **WinMLS** [53]: software para medidas de calidad de audio, acústicas y vibracionales. Calcula además de parámetros acústicos, respuestas al impulso de recintos a través del método [MLS](#).

### 3.2.2 Comparativa

Una vez vista una descripción de algunos sistemas de simulación acústica, se detalla en este apartado algunas características concretas de cada uno de ellos que pueden servir a la hora de la elección de una u otra herramienta. En la Tabla 3.1 se muestra una comparación de las herramientas en cuanto a la teoría de simulación acústica que utilizan. Además, se incluye si el programa realiza un cálculo de parámetros acústicos y está preparado para realizar auralización.

	Teoría geométrica	Teoría ondulatoria	Parámetros acústicos	Auralización
<b>ABEC3</b>		BEM,LEM		
<b>AIST-Laboratory</b>	Trazado de rayos		✓	
<b>Aurora</b>			✓	✓
<b>Cara</b>	Fuente imagen		✓	✓
<b>CATT-Acoustic</b>	Fuente imagen, trazado de conos		✓	✓
<b>EASE</b>	Trazado de rayos		✓	
<b>LMS RAYNOISE</b>	Trazado de rayos		✓	✓
<b>ODEON</b>	Fuente imagen, trazado de rayos		✓	✓
<b>Olive Tree Lab - Room</b>	Trazado de rayos		✓	✓
<b>RAMSETE</b>	Trazado piramidal		✓	✓

Tabla 3.1: Comparativa funcional de sistemas de simulación comerciales

Por su parte, en la Tabla 3.2 se realiza una comparación de los sistemas de simulación indicando la licencia disponible y el sistema operativo donde puede ejecutarse. Para más información del tipo de licencia véase Anexo A.

	Tipo de licencia	Windows	Linux	Mac OS X
<b>ABEC3</b>	Demo y estudiante	✓		
<b>AIST-Laboratory</b>	Demo	✓		
<b>Aurora</b>	Paquete libre sobre Adobe Audition	✓		✓
<b>Cara</b>		✓		
<b>CATT-Acoustic</b>	Demo	✓		
<b>EASE</b>		✓		
<b>LMS RAYNOISE</b>		✓	✓	
<b>ODEON</b>	Demo	✓		
<b>Olive Tree Lab - Room</b>	Demo	✓		
<b>RAMSETE</b>	Demo	✓	✓	✓
<b>RoomOptimizer</b>	Shareware	✓		
<b>RPlusD</b>	Demo	✓		
<b>WinMLS</b>	Demo	✓		

Tabla 3.2: Comparativa de sistemas de simulación con licencia de software propietario

La mayor parte de los sistemas de simulación de este apartado simulan en 3D, a excepción de RoomOptimizer, RPlusD y WinMLS. Cabe destacar que RoomOptimizer únicamente acepta recintos con forma cuboide y por su parte, AIST-Laboratory, Cara y CATT-Acoustic permiten la simulación de recintos con geometrías arbitrarias.

Destacan de sistemas de simulación acústicos comerciales las herramientas CATT-Acoustic y EA-SE, ambos ampliamente utilizados en el mundo de la educación e investigación, así como en el ámbito profesional.

### 3.3 Sistemas libres

En esta sección se describen aquellas herramientas de código abierto, o según se ha explicado en el Anexo A, también denominadas software libre. Como se vio en los objetivos de este TFM, se trata de encontrar aquella herramienta de software libre que reúna el mayor número de características deseadas, para que pueda ser utilizada en el sistema de simulación desarrollado.

#### 3.3.1 Breve descripción

A continuación se realiza una breve descripción de cada una de las herramientas de este tipo que tienen interés:

1. **Abersim** [54]: se trata de un conjunto de herramientas para simular la propagación de ondas acústicas no lineales en 3D en un medio con atenuación. Enfocado principalmente al campo de la imagen médica por ultrasonido, aunque puede ser aplicado en otras áreas. Dispone de una interfaz en Matlab donde se puede visualizar y procesar la información. No obstante, está implementado en código C para aprovechar la computación en paralelo *Message Passing Interface* (MPI).
2. **AC2D** (Acoustic FDTD Solver) [55]: herramienta para simular la propagación de ondas acústicas por el método **FDTD** en medios con pérdidas y 2D. El formato de entrada es XML y el de salida H5 al tener que procesar gran cantidad de datos, realizándose la visualización de resultados a través de Matlab. Se puede usar para determinar los coeficientes de reflexión y transmisión del medio, así como para resolver la ecuación de onda dispersada por objetos acústicos debido a una onda plana o una onda isotropa.
3. **AcMus** [56]: programa para investigación de recintos acústicos que consta de herramientas para medición, análisis y simulación de recintos destinados a la producción y reproducción de música. A través de ordenadores portátiles, altavoces y micrófonos, esta herramienta evalúa los parámetros acústicos relevantes para determinar los problemas que se puedan encontrar en la sala. La simulación se basa en métodos basados en la teoría geométrica, como son el método de fuente imagen y trazado de rayos.
4. **Acoustic Simulator** [57]: se trata de un proyecto que realiza una simulación de la propagación del sonido en una escena 3D utilizando la técnica de la teoría ondulatoria **FDTD**. Está desarrollado en C++ y para su construcción y ejecución se requiere Visual Studio 2015 o posterior.
5. **AcouSTO** (Acoustic Simulation Tool) [58]: simulador basado en el método **BEM** de la teoría ondulatoria. Aprovecha simetrías para reducir el tiempo de computación, dispone de ejecución en paralelo MPI2, acepta geometrías arbitrarias en 3D y entre otras características, es posible señalar un número arbitrario de fuentes. Se trata de una herramienta potente que incluye una interfaz a MySQL para almacenar datos fácilmente.

6. **BEM++** [59]: es una librería de elementos de contorno Galerkin que se encarga de resolver ecuaciones de Laplace, Helmholtz y Maxwell en dominios acotados y no acotados.
7. **BEM Acoustics** [60]: es una herramienta de Matlab basada en el método **BEM** en el dominio temporal que calcula problemas de radiación y dispersión de sonido.
8. **Blender** [61]: se trata de una herramienta para modelado, iluminación, renderizado, animación y creación de gráficos tridimensionales. Desarrollado en Python, tiene una gran popularidad y en la parte de acústica realiza la técnica geométrica de trazado de rayos.
9. **CABRILLO** [62]: se trata de un código que utiliza el método pseudo-espectral *Finite-Difference* (FD) de la teoría ondulatoria para acústica en medios elásticos y poroelásticos. La ventaja de la técnica **FD** es que es posible modelar entornos con geometrías complejas. Al ser un método pseudo-espectral se requiere escasa memoria y bajo número de cálculos para conseguir una precisión adecuada. Como inconveniente se encuentra que no es posible modelar la atenuación, aspecto conseguido con la teoría de Biot.
10. **Clawpack** (Conservation Laws Package) [63]: es una colección de métodos para volúmenes finitos para sistemas hiperbólicos lineales y no lineales de conservación de leyes. Emplea métodos de alta resolución de tipo Godunov aplicable a muchos tipos de onda. Usa ecuaciones Riemann para dar las soluciones generales.
11. **CREANIUS** [64]: este software permite simular imágenes de radio frecuencia de ultrasonidos no lineales obtenidas por fuentes lineales. Dispone de dos enfoques diferentes: un simulador de campo de propagación no lineal basado en un método de espectro angular generalizado, y un esquema de reconstrucción de imagen usando campos no lineales.
12. **Code TYMPAN** [65]: es un software de propagación acústica en escenas 3D complejas, destinado a la evaluación y predicción del impacto del ruido en entornos industriales. Desarrollado para permitir estudios de ingeniería acústica.
13. **DREAM** (Discrete REpresentation Array Modelling) [66]: se trata de una herramienta para Matlab y/o Octave para simular campos acústicos radiados desde transductores ultrasónicos comunes y arrays complejos. Permite, entre otros, analizar la dirección y enfoque del haz tanto para campos cercanos como lejanos. Se puede usar para simular y evaluar diseños de arrays o evaluar métodos de imagen.
14. **Edge Diffraction Toolbox** [67]: es una herramienta realizada en Matlab para calcular reflexiones especulares y difracciones de un alto orden, usando formulaciones en el dominio temporal o frecuencial. Usa el método de fuente imagen de la teoría geométrica y varios métodos disponibles para calcular las respuestas al impulso de las difracciones.
15. **ElmerFEM** [68]: es un software de simulación multifísica que incluye modelos de fluidos dinámicos, mecánica estructural, electromagnetismo, transferencia de calor y acústica, entre otros. Estos modelos se describen por ecuaciones diferenciales parciales por el método **FEM**.
16. **EVERTims** [69]: forma parte del proyecto EVERT, el cual se divide en varias partes funcionando individualmente como módulos cada una de ellas, a destacar:
  - EVERT: módulo que realiza el método de trazado de rayos.
  - EVERTims: realiza simulación acústica de recintos basándose en el módulo EVERT.

- Existen otros dos módulos dedicados a la visualización a través de Blender y otro dedicado a la auralización.
17. **FDAC3DMOD** [70]: es una herramienta de simulación acústica 3D que resuelve ecuaciones de velocidad-presión por el método **FDTD** de la teoría ondulatoria usando como condición de contorno *Perfectly Matched Layers* (PML).
  18. **Image-Sources** [71]: se trata de un ejercicio propuesto dentro de la asignatura Geometría Digital 3D de la Universidad Duke. Se propone representar las reflexiones especulares en un entorno simple para simular la respuesta al impulso de una fuente y receptor en un escenario 3D. El método a utilizar es fuente imagen de la teoría geométrica, con lenguaje Javascript.
  19. **i-Simpa** [72]: plataforma dedicada a modelar la propagación del sonido en dominios complejos en 3D. Destinada a expertos en el mundo de la acústica, entornos académicos o investigadores. Se puede aplicar a recintos acústicos, entornos urbanos o espacios industriales, entre otros. Aunque en principio realiza el método de trazado de rayos, trazado de sonido-partícula y teoría de la reverberación, también utiliza enfoques de teoría ondulatoria. Por sí solo no es un software de cálculo, pero es equivalente a un pre y post procesador para códigos acústicos.
  20. **k-Wave** [73]: herramienta acústica para Matlab, diseñada para simulaciones acústicas en el dominio temporal y simulaciones de ultrasonidos. Se centra en la propagación de ondas acústicas en medios heterogéneos y tiene una gran potencia de cálculo.
  21. **KZK Bergen Code** [74]: resuelve la ecuación *Khokhlov-Zabolotskaya-Kuznetsov* (KZK) en el dominio frecuencial en 1D y 2D. Desarrollada en lenguaje de programación Fortran.
  22. **KZK Texas Code** [75]: resuelve a través de la ecuación **KZK** en el dominio temporal los haces sonoros con simetría axial en fluidos. Desarrollada en Fortran y C++.
  23. **MIMPy** [76]: entorno para simulaciones acústicas simples en recintos rectangulares. Calcula la respuesta al impulso a partir de la colocación de fuentes y receptores ideales. Es posible la visualización del campo sonoro del recinto. El objetivo del proyecto es el cálculo de las respuestas impulsivas binaurales del recinto rectangular.
  24. **NiHu** [77]: es una herramienta de Matlab desarrollada en C++ usada para resolver problemas de contorno por medio de ecuaciones diferenciales parciales por el método **BEM**. Surge para unificar problemas **BEM** de diferentes tipos.
  25. **Open Acoustics** [78]: se trata de un proyecto en construcción para simulación acústica desarrollado en Python. Liderado en la actualidad por Takuya OSHIMA.
  26. **OpenPSTD** [79]: herramienta que permite calcular eficientemente y con detalle la propagación del sonido en entornos cerrados. Utiliza el método ondulatorio *Pseudo-Spectral Time-Domain* (PSTD).
  27. **Pachyderm Acoustical Simulation** [80,81]: es una colección de algoritmos de simulación acústica que pueden ser usados para predicción de ruido o visualización de la propagación del sonido. Otras capacidades de la herramienta son: cálculo de coeficientes de absorción, implementación de **FDTD**, difracción en bordes y auralización.
  28. **Parallel FDTD** [82]: resuelve por el método ondulatorio **FDTD** simulación acústica de recintos usando lenguaje CUDA para acelerar los cálculos.



29. **Pogo** [83]: resuelve problemas elastodinámicos usando el método **FDTD**. Se ejecuta en tarjetas gráficas Nvidia permitiendo de este modo mejorar las velocidades de simulación. Resuelve ecuaciones de onda elástica en 2D y 3D.
30. **RoomAcoustics** [84]: programa que determina la cantidad de reverberación existente en un recinto. Introduciendo las dimensiones de la sala, las ventanas y puertas, así como los materiales de las superficies del suelo, paredes y techo, es capaz de calcular los modos gráficamente.
31. **Room Acoustic Simulation with CUDA** [85]: sistema de simulación acústica de recintos 2D. Es capaz de simular el sonido para un oyente en una posición designada en el mapa del recinto dado con múltiples fuentes de sonido. El mapa del recinto 2D y las posiciones de las fuentes de sonido necesitan ser designadas en el proceso de presimulación. El sistema permite cambiar la posición del oyente y de las fuentes en tiempo real.
32. **Roomsim** [86]: simulador acústico de un recinto rectangular desarrollado en Matlab. Hace uso del método geométrico fuente imagen y calcula algunos efectos acústicos.
33. **SimSonic** [87]: es un software de simulación de propagación de ultrasonidos basada en el método **FDTD**. No obstante, es una herramienta genérica que se puede aplicar a distintas situaciones de propagación de ultrasonidos. Existe una versión de cálculos en 2D y otra en 3D. Se trata de un conjunto de programas compilados y códigos fuente C, con herramienta de Matlab para gestionar la entrada y salida de datos.
34. **Slab3d** [88]: es un sistema de renderizado de entornos virtuales en tiempo real desarrollado originalmente en el Spatial Auditory Displays Lab en el Centro de Investigación Ames de la NASA. Es capaz de realizar el procesamiento de sonido en un entorno espacial 3D, permitiendo la asignación arbitraria de las fuentes de sonido en el espacio auditivo. Su anterior versión fue la conocida herramienta SLAB.
35. **Ultrasim** [89]: se trata de una herramienta de Matlab cuya última actualización se encuentra en [90]. Es un simulador que dispone de un menú interactivo orientado a encontrar el campo sonoro desde transductores, principalmente desde ultrasonidos médicos, pero que también puede ser usado por sónar e incluso altavoces.
36. **Wave Cloud** [91]: se trata de un simulador bastante potente basado en el método de ondas **FDTD**. Justifican el uso de este método por el compromiso entre la precisión y la eficiencia computacional. Es capaz de dar solución a comportamientos transitorios y variables en el tiempo, por ejemplo, responde ante fuentes móviles, receptores y entorno en movimiento. Wave Cloud ha sido diseñado pensando en la aplicación a acústica de recintos, por este motivo, se ha teniendo en cuenta la posibilidad de recintos con geometrías arbitrarias o el modelado de fuentes y receptores direccionales. Además, se presta atención a la visualización 3D del sonido, así como a las tan conocidas actualmente estrategias de auralización.

### 3.3.2 Comparativa

Una vez se ha realizado una breve descripción de cada una de las herramientas de software libre con algún tipo de relevancia para este trabajo, se procede a resumir en la Tabla 3.3 las principales características de cada una de ellas para realizar una selección de forma más sencilla.

### 3.3.3 Selección

Una vez se dispone de las principales características de cada herramienta en la Tabla 3.3, se seleccionan aquellas herramientas que cumplan los siguientes requisitos:

- La licencia de software libre permita disponer del código fuente y su modificación.
- La geometría del recinto a simular no esté restringida a formas cuboides o paralelepípedas y se permita la definición de obstáculos.

Las herramientas que según la definición del autor cumplen estas condiciones y por tanto se seleccionan en una primera fase son:

- AcMus
- Acoustic Simulator
- AcouSTO
- Edge Diffraction Toolbox
- Elmer
- EVERTims
- Image-Sources
- Pachyderm Acoustical Simulation
- Parallel FDTD
- Wave Cloud

Además, cabe destacar que se prefiere que la herramienta de simulación cumpla los siguientes aspectos:

- El lenguaje de programación sea C, C++, CUDA o Matlab, con preferencia del último.
- La simulación se realice con teoría geométrica.

Por tanto, las herramientas que cumplen estas especificaciones son Edge Diffraction Toolbox y EVERTims, la primera desarrollada en Matlab y la segunda en C++. Se realiza una prueba de ambas, no llegando con la segunda a buenos términos, por tanto, dado que la primera está desarrollada en Matlab y se trata de un lenguaje sencillo, se selecciona esta.

## 3.4 Edge Diffraction Toolbox

Según la descripción de su sitio web [67], la herramienta para Matlab *Edge Diffraction Toolbox* (EDT), es capaz de calcular las reflexiones especulares y altos órdenes de difracción usando formulaciones en el dominio temporal o frecuencial. Se trata de una formulación iterativa, es decir, se calcula un orden de difracción por tiempo, aunque este aspecto en la última versión de la herramienta se intenta mejorar. Únicamente acepta superficies rígidas ideales (Neumann) y condiciones Dirichlet de presión.

	Licencia	Lenguaje de programación	Teoría geométrica	Teoría on-duladora	Simulación 3D	Windows	Linux	Mac OS
Abersim	GNU GPL	C, Matlab			✓	✓	✓	✓
AC2D	BSD	C++,Matlab		FDTD			✓	✓
AcMus	GNU GPL	Java	Fuente imagen,trazado de rayos		✓	✓	✓	✓
Acoustic Simulator	GNU GPL	C++,Matlab		FDTD	✓	✓	✓	✓
AcouSTO	GNU GPL	C		BEM	✓	✓	✓	✓
BEM++	BSD	C++,Python		BEM		✓	✓	✓
BEM Acoustics	GNU GPL	Matlab		BEM		✓	✓	✓
Blender	GNU GPL	Python	Trazado de rayos		✓	✓	✓	✓
CABRILLO	Contactar	Fortran,Matlab		FDTD		✓	✓	✓
Clawpack	BSD	Python,Fortran		FEM		Cygwin	✓	✓
CREANIUS	CeCILL-B	C,C++		✓		✓	Fedora	
Code TYMPAN			Trazado de rayos		✓			
DREAM	GNU GPL	Matlab,Octave				✓	✓	✓
Edge Diffraction Toolbox	GNU GPL	Matlab	Fuente imagen		✓	✓	✓	✓
ElmerFEM	GNU GPL	C		FEM	✓	✓	✓	
EVERTMS		C++	Trazado de rayos, trazado de haz		✓	✓	✓	✓
FDAC3DMOD		C		FDTD	✓		✓	
Image-Sources	GNU GPL	Javascript	Fuente imagen		✓	✓	✓	✓
i-Simpa	GNU GPL	C++,Python	Trazado de rayo		✓	✓		
k-Wave	LGPL	Matlab,C++		✓	✓	✓	✓	
KZK Bergen Code		Fortran		KZK				
KZK Texas Code		Fortran,C++		KZK				
MIMPY	GNU GPL	Python,C	Fuente imagen			✓	✓	✓
NiHu	GNU GPL	C++,Matlab		BEM	✓	✓	✓	
OpenPSTD		Python,Blender,Numpy		PSTD	✓	✓	✓	✓
Pachyderm A. S.	GNU GPL	.NET	Trazado de rayos			✓		
Parallel FDTD	GNU GPL	CUDA,Python,Matlab		FDTD		✓	✓	
Pogo	GNU GPL	CUDA,C++		FEM	✓	✓	✓	✓
RoomAcoustics	GNU GPL	Java						
R. A. S. with CUDA	GNU GPL	C++,CUDA		FDTD	✓	✓		
Roomsim	GNU GPL	Matlab	Fuente imagen		✓	✓	✓	✓
SimSonic	GNU GPL	C,Matlab		FDTD	✓	✓	✓	
Slab3d	NASA Open Source	C++,Matlab			✓	✓		
Ultrasm	GNU GPL	Matlab				✓	✓	✓
Wave Cloud	GNU GPL	C,CUDA,Matlab		FDTD		✓		

Tabla 3.3: Características herramientas de simulación con licencia libre

### 3.4.1 Autor

El autor principal de esta herramienta es Peter Svensson [92]. Es profesor de electroacústica en el Departamento de Electrónica y Telecomunicaciones de la Universidad Noruega de Ciencia y Tecnología (NTNU) ubicada en Trondheim (Noruega). Además, es miembro de la Sociedad Americana de Acústica, miembro de las Sociedades Acústicas de Noruega, Suecia y Japón, así como de la Sociedad de Ingeniería de Audio, y del IEEE. A lo largo de los años ha poseído distintos cargos en la Sociedad de Suecia, Noruega y en la Asociación Europea de Acústica.

### 3.4.2 Licencia

La herramienta EDT es un software libre, se puede redistribuir y/o modificar bajo los términos de la licencia GNU GPL según publica la Free Software Foundation, ya sea en su versión 3 de la licencia, o en cualquier versión posterior. EDT es distribuido por su autor con la esperanza de que sea útil, pero sin aportar ninguna garantía. Para más información sobre esta licencia ver Anexo A. Cabe resalta que la herramienta dispone de derechos de autor fechados en 1999-2012 por Peter Svensson, de la Universidad Noruega de Ciencia y Tecnología.

### 3.4.3 Versiones de la herramienta

En la actualidad, están disponibles 4 versiones del software EDT, la última fechada a 30 de junio de 2015. Estas son:

- Versión B0: realiza simulación en el dominio temporal con una formulación iterativa. Su última modificación data de febrero de 2010, constituyendo así la primera versión de EDT.
- Versión B1: realiza simulación en el dominio temporal con una formulación iterativa, al igual que la versión B0. Su última modificación data de abril de 2012.
- Versión B2: realiza simulación en el dominio temporal y frecuencial con una formulación iterativa. Su última modificación data de abril de 2015.
- Versión ESIE0: realiza simulación en el dominio frecuencial basándose en la resolución de ecuaciones diferenciales. En su última versión (junio 2015) se trata de una versión preliminar.

A la vista de estas versiones, se selecciona la versión B2 para estudiarla y aplicarla en el desarrollo del sistema de simulación que constituye este TFM ya que dispone de una fecha de modificación más reciente.

### 3.4.4 EDT versión B2

#### 3.4.4.1 Descripción

Conocida como EDB2, esta herramienta contiene un conjunto de funciones de Matlab que calculan la respuesta al impulso (IR), y/o la función de transferencia (*Transfer Function* (TF)), para una o varias fuentes puntuales en un entorno de superficies planas y rígidas en la que se pueden indicar varios receptores. Se incluyen reflexiones especulares (hasta orden 5-10, dependiendo del número de planos y la capacidad de memoria del ordenador donde se ejecute el programa) y difracciones en bordes (hasta orden 6 para IR, y hasta segundo orden para TF), también se incluyen combinaciones de las anteriores.

Un único programa, denominado EDB2main, hace todos los cálculos pero crea archivos intermedios de resultados de modo que se convierta en una herramienta modulable. Como se ha adelantado, se pueden especificar varias fuentes y receptores, ejecutándose cada una como un trabajo en *batch*, y generando un archivo de salida para cada par fuente-receptor.

### 3.4.4.2 Método de cálculo

El método de cálculo se puede dividir en dos pasos principales:

1. Si al comenzar se señala la variable `findpaths=1`, se buscan las trayectorias de los rayos válidos de reflexión y difracción. Para ello se aplica el método de fuente imagen [28] con extensiones para difracciones en bordes [18].
2. En este segundo paso es posible aportar medidas en el dominio temporal y/o frecuencial, existiendo dos opciones diferenciadas:
  - a Para calcular la respuesta al impulso se señala la variable `calctirs` a nivel 1. Cada trayectoria válida calculada en el paso 1 genera una respuesta al impulso diferente que se añade a la total. Se distinguen las IR de las reflexiones especulares (un pulso), de las IR de las difracciones que usan la expresión especificada en [19].
  - b Para calcular la función de transferencia se señala la variable `calctfs` a nivel 1. Cada trayectoria genera una, siendo añadidas todas ellas a la TF total.

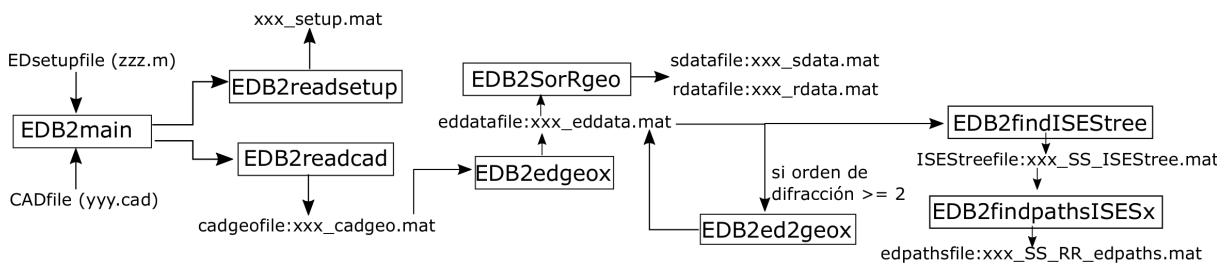


Figura 3.1: Esquema de cálculo de la herramienta EDT

En la Figura 3.1 se muestra un esquema de los distintos pasos seguidos por la herramienta para la obtención de los resultados de la simulación. De dicho esquema se destacan dos aspectos necesarios para comenzar con la ejecución de la herramienta:

1. Es necesario disponer de un archivo con la geometría del recinto u obstáculo a simular en formato `.cad` (generado por el software CATT-Acoustic [44]) y denominado en el esquema como `CADfile`.
2. También es necesario disponer de un archivo de configuración con los valores de los distintos parámetros de cálculo denominado `EDsetupfile`.

Se puede comprobar como existen multitud de archivos intermedios con resultados con el objetivo de que dichos módulos puedan ser empleados libremente en otra herramienta de forma sencilla, siempre y cuando se generen archivos con las variables correctas para que todo funcione correctamente.

A continuación se realiza una descripción de cada una de las funciones que forman la herramienta. En el esquema de la Figura 3.1 se han representado las funciones principales, no obstante, esas funciones realizan llamadas a otras funciones que se desgranarán a continuación:

- `EDB2betaoverml`, `EDB2betaoverml2_fd`, `EDB2betaoverml_fd`: se utilizan para realizar operaciones con mayor rapidez, la primera calcula [IR](#) de difracciones en bordes y las restantes [TF](#).
- `EDB2calcdist`: devuelve la distancia entre un conjunto de puntos definidos en un espacio en 3D.
- `EDB2checkobstr_edgetoedge`: comprueba las obstrucciones para una lista de rayos cuya trayectoria va de un borde a otro borde comprobando si los rayos pasan a través de cualquiera de los planos, en ese caso existiría una obstrucción.
- `EDB2checkobstr_pointtoedge`: comprueba las obstrucciones para una lista de rayos cuya trayectoria va desde puntos a bordes comprobando si los rayos pasan a través de cualquiera de los planos, en cuyo caso existiría una obstrucción.
- `EDB2checkobstrpaths`: comprueba las obstrucciones para una lista de rayos, o comprueba si las reflexiones especulares son válidas comprobando que no atraviesen ningún plano ya que en ese caso existiría una obstrucción.
- `EDB2chkISvisible`, `EDB2chkISvisiblex`: comprueba si los rayos desde un conjunto de fuentes imagen a un conjunto de receptores pasan a través de los planos reflejantes.
- `EDB2compress7`: comprueba para combinaciones de bordes a bordes que tengan idénticos valores.
- `EDB2coordtrans1`, `EDB2coordtrans2`: transforma uno o dos conjuntos de coordenadas cartesianas a coordenadas cilíndricas.
- `EDB2creindeixmatrix`: crea una matriz de índices.
- `EDB2cross`: cálculo del producto cruzado de forma más simplificada que la función propia de Matlab.
- `EDB2diff2ISES`: da una lista de trayectos que incluyen una combinación de orden 2 de difracción, es decir, trayectorias que incluyen un rayo difractado de segundo orden y posiblemente reflexiones especulares antes y después.
- `EDB2diff1SESx`: da una lista de rayos que incluyen una combinación de orden 1 de difracción, es decir, trayectorias que incluyen rayos de difracción de un orden y posiblemente con reflexiones especulares antes y después.
- `EDB2diffNISES`: da una lista de rayos que incluyen un rayo de difracción de orden N, es decir, aporta una lista de rayos visibles de difracción N, posiblemente con reflexiones especulares antes y después.
- `EDB2directsound`: comprueba si el sonido directo (desde la fuente al receptor) es válido comprobando si existe obstrucción debida a cualquier plano.
- `EDB2ed2geox`: calcula los parámetros geométricos relativos a bordes y planos de segundo y mayor orden basándose en vértices y planos definidos en un archivo `.mat` generado por la función `EDB2readcad`. Estos parámetros calculados son independientes de las posiciones de fuentes y receptores.
- `EDB2edgeox`: calcula los parámetros geométricos relativos a planos y bordes basándose en vértices y planos definidos en un archivo `.mat` generado por la función `EDB2readcad`. Estos parámetros son independientes de las posiciones de fuentes y receptores.

- `EDB2editpathlist`: realiza una edición semiautomática del archivo de salida `edpathsfile`, aumentando, conservando o eliminando rayos calculados en un primer momento.
  - `EDB2extrnums`: extrae el valor numérico de un texto en función de un delimitador.
  - `EDB2findis`: devuelve las coordenadas de las fuentes imagen de una o más fuentes y de una lista de planos.
  - `EDB2findISEStree`: construye un árbol con todas las posibles combinaciones de reflexiones especulares y difracciones para una fuente dada. Se basa en las matrices de visibilidad desde la fuente a los planos y bordes y entre los planos y bordes. Las comprobaciones de visibilidad se emplean si es posible y no se comprueban las que dependen de la posición del receptor.
  - `EDB2findpathsISESx`: encuentra todos los posibles trayectos incluyendo al sonido directo, reflexión especular y difracción.
  - `EDB2fistr`: lee un archivo de texto línea por línea hasta que encuentra una determinada cadena de caracteres.
  - `EDB2getedgepoints`: calcula las coordenadas de un borde.
  - `EDB2infrontofplane`: comprueba la posición de un punto (si está en frente, detrás o en el plano). También comprueba si un único plano esta en una lista de puntos.
  - `EDB2irfromslotlist`: calcula una [IR](#) a partir de una lista de valores de muestreo y amplitudes.
  - `EDB2main`: es el archivo principal de la herramienta, encargada de calcular las reflexiones especulares y difracciones de bordes para obtener la respuesta al impulso del recinto. Para cada fuente y receptor genera un archivo diferente. Los tres pasos de cálculo principales son:
    1. Cálculos geométricos: define los bordes y determina la visibilidad entre los planos, bordes, fuentes y receptores.
    2. Busca los rayos sonoros válidos.
    3. Construye a partir de los rayos sonoros del paso anterior la respuesta al impulso.
- Incluye la opción de dividir los cálculos en batches ya que cuando existen muchos receptores los precálculos pueden sobrecargar la memoria si todos los receptores se tratan a la vez para cada fuente.
- `EDB2makeirs`, `EDB2maketfs`: construye las respuestas al impulso y las funciones de transferencia desde una lista de trayectorias.
  - `EDB2makemovie`, `EDB2makesnapshot`: realiza una película y una fotografía respectivamente de la respuesta al impulso.
  - `EDB2myvalueinput`: escribe un texto en pantalla y espera a que se introduzca un valor numérico comprobando si está dentro de un rango.
  - `EDB2plotmodel`: dibuja un modelo que es dado por el archivo `eddatafile`.
  - `EDB2plotpath`: dibuja un modelo que es dado por el archivo `eddatafile` y uno o más rayos.
  - `EDB2plotstackedIRs`, `EDB2plotstackedTFs`: representan un número de [IR](#)/[TF](#) de forma apilada.

- `EDB2poinpla`, `EDB2poinplax`: detecta si uno o más puntos están dentro de un número finito de planos.
- `EDB2readac`: lee un archivo tipo de `.ac` (realizado por ejemplo con Invis AC3D) y salva toda la geometría en un archivo `.mat`.
- `EDB2readcad`: lee un archivo de tipo `.cad` (realizado por ejemplo con CATT-Acoustic) y salva todos los datos geométricos en un archivo `.mat`. Se leen archivos `.cad` v6, v7 y v8.
- `EDB2readsetup`: ejecuta el archivo `EDsetup` y salva todas las configuraciones en un archivo `.mat`. Si las fuentes y receptores se toman desde el archivo `.cad`, serán extraídos y salvados en el archivo `setupmatfile`.
- `EDB2SorRgeo`: calcula una serie de parámetros geométricos relativos a fuentes o receptores basándose en vértices, bordes y planos así como en una lista de coordenadas de fuentes/receptores.
- `EDB2speculISES`: encuentra las reflexiones especulares válidas comprobando la visibilidad y obstrucción, dando una lista de potenciales fuentes imagen.
- `EDB2strpend`: elimina la extensión de un archivo.
- `EDB2wedge1st_int`, `EDB2wedge2nd`, `EDB2wedgeN`: da la respuesta al impulso de primer, segundo y N orden de difracción.
- `EDB2wedge1st_fd`, `EDB2wedge2nd_fd`: da la función de transferencia para orden 1 y 2.

#### 3.4.4.3 Tiempo de cálculo y uso de memoria

La herramienta [EDT](#) utiliza vectorización en un alto grado para acelerar el procesamiento, aunque esto trae consigo la existencia de matrices de tamaños elevados como variables intermedias, especialmente en las funciones `EDB2SorRgeo`, `EDB2ed2geox` y `EDB2findISEStree`. En la primera de las funciones es posible limitar la vectorización introduciendo bucles `for` pero para las otras dos no hay soluciones sencillas. En consecuencia, para reflexiones especulares de orden mayor de 6, a menudo conducen a mensajes de error de fuera de memoria incluso para modelos pequeños.

El archivo generado en la función `EDB2findISEStree` (`ISEStreefile`) contiene todas las fuentes imágenes válidas y bordes-fuente (esto es, los bordes que están en el lado derecho del ultimo plano de reflexión), pero la comprobación de visibilidad (es decir, si los puntos de impacto están dentro de planos finitos) y la comprobación de obstrucción no se realiza hasta que se ejecuta la función `EDB2findpathsISESx`. Por tanto, el `ISEStree` es muy grande, y algunas matrices no se transfieren a través de un archivo entre las funciones `EDB2findISEStree` y `EDB2findpathsISESx`, sino a través de variables globales. El uso del algoritmo de trazado de haz haría que estos pasos fueran mucho mas eficientes que la presente aplicación debido a que el `ISEStree` podría reducirse mucho. El tiempo de cálculo de las respuestas al impulso depende en gran medida de la frecuencia de muestreo seleccionada. Esta herramienta también dispone de una versión compilada disponible para Mac OS X para la cual la generación de [IR](#) de difracción de primer orden es dos veces mas rápida que la versión de Matlab.

## 3.5 Otras herramientas de interés

Además de los sistemas especificados en los anteriores apartados, se destacan en este apartado dos herramientas con licencia de software libre que aunque no realizan simulación acústica pueden ser de ayuda en proyectos de este tipo y que merecen ser mencionadas [\[93\]](#):



1. **AlphaDoc:** se trata de una base de datos con interfaz gráfica que recoge los coeficientes de absorción de distintos materiales de construcción.
2. **VizIR:** es una herramienta gráfica que permite comparar fácilmente distintos parámetros acústicos como puede ser por ejemplo el tiempo de reverberación. Es capaz de importar resultados desde herramientas como CATT-Acoustic o Aurora. También calcula automáticamente parámetros acústicos desde una respuesta al impulso que le sea introducida.

## 3.6 Conclusiones

Existen multitud de herramientas de simulación, tanto de software libre como propietario, no obstante, tras este estudio se puede resaltar el hecho que los programas propietarios cuentan con un potencial bastante superior a las herramientas libres. Aunque cada vez el número de herramientas libres supera al de comerciales, en parte por los intentos de los investigadores acústicos de disponer de sus propias herramientas de simulación y no depender de terceros.

Tras una ardua labor de investigación de herramientas de simulación existentes y de comprobación de sus características, se ha procedido a la selección de aquella que cumple la mayor parte de requisitos impuestos y se cree que ayudará en el desarrollo del sistema de simulación a implementar en este [TFM](#).



# Capítulo 4

## Desarrollo

### 4.1 Introducción

En este capítulo se especifican aquellos desarrollos necesarios para crear un sistema de simulación acústico en entornos cerrados tratando de utilizar la herramienta ya desarrollada de software libre [EDT](#) explicada en el Capítulo 3.

Por un lado se ha creado una aplicación que incluye una interfaz gráfica de usuario (*Graphical User Interface* (GUI)) desarrollada en Matlab con el entorno de desarrollo de [GUI](#) [94]. Esta aplicación permite al usuario del sistema de simulación controlar la aplicación de forma sencilla, sin necesidad de conocer el lenguaje de programación. Por otro lado, se ha creado una interfaz de programación de aplicaciones, más conocida como *Application Programming Interface* (API), que contiene un conjunto de funciones y procedimientos para ser utilizados por otro software como una capa de abstracción. En el primer desarrollo se obtiene una visualización gráfica de los resultados de simulación, mientras que en el segundo se obtiene una matriz y estructura de Matlab con los resultados.

El presente capítulo se estructura en cinco secciones principales:

1. En la primera se incluyen los objetivos que se buscan en el sistema de simulación a desarrollar.
2. En la segunda se explican aquellos desarrollos realizados en la etapa de presimulación, aspectos necesarios para poder utilizar la herramienta [EDT](#).
3. En la tercera sección se incluyen los aspectos más relevantes para ejecutar la simulación.
4. En la cuarta sección se incluye el desarrollo del sistema de visualización y de obtención de resultados.
5. En la quinta sección se especifican aquellos aspectos a tener en cuenta a la hora de utilizar la herramienta.

En las secciones donde es necesario se realiza una subdivisión de la misma diferenciando la interfaz gráfica de usuario y la interfaz de programación de aplicaciones, poniendo de relieve los aspectos necesarios en cada una de ellas.

Se recomienda consultar el Apéndice [B](#) para conocer la forma de utilización tanto de la interfaz gráfica de usuario como de la [API](#).

## 4.2 Objetivos del sistema de simulación

Antes de comenzar con los desarrollos realizados, se especifican aquellos aspectos que tienen que ser considerados en el sistema de simulación a desarrollar:

- Crear una interfaz gráfica de usuario.
- Crear una interfaz de programación de aplicaciones.
- Simular recintos con obstáculos definidos en formato `.env/.srf` y `.cad`.
- Ser capaz de simular recintos con geometría no paralelepípeda.
- Seleccionar la posición de fuentes y receptores.
- Seleccionar parámetros de la simulación.
- Obtener las trayectorias de los rayos acústicos.

## 4.3 Preprocesamiento

Como se comprueba en la Figura 3.1 del capítulo 3, para realizar la llamada a la función `EDB2main` que da acceso al comienzo de la simulación de la herramienta [EDT](#) es necesario disponer de un archivo con la geometría en formato `.cad` y un archivo de configuración en formato Matlab.

### 4.3.1 Interfaz gráfica

La interfaz gráfica de usuario en la etapa anterior a la simulación consta de dos interfaces:

1. Interfaz de bienvenida al sistema de simulación con nombre de archivo `simulador1` con la apariencia mostrada en la Figura 4.1.
2. Interfaz de introducción de parámetros e información de simulación con nombre de archivo `simulador2` con la apariencia mostrada en la Figura 4.2.

En la primera de las interfaces no tiene lugar ningún preprocesamiento de datos, no obstante, en la segunda de las interfaces tienen lugar varias etapas de procesamiento para obtener los datos necesarios y ejecutar la simulación. En los siguientes subapartados se detallan estas etapas.

#### 4.3.1.1 Conversión del formato de archivos de la geometría

Uno de los objetivos del sistema de simulación a desarrollar es que sea capaz de aceptar como formato geométrico archivos con extensión `.env/.srf`. Por tanto, es necesario realizar una conversión de esta extensión a `.cad`, formato aceptado por la herramienta [EDT](#).

El formato de archivos `.env/.srf` especifica la geometría de un recinto y diversos obstáculos definiendo en un único archivo `.env` de nombre normalmente `environment`, el número de superficies que componen toda la geometría, así como los archivos `.srf` que contienen a cada una de las superficies. Además, el archivo `.env` incluye otra información como es un comentario del recinto que se está definiendo, entre otras. En el archivo `.srf` se especifica el número de vértices de la superficie que se está definiendo y las coordenadas  $x, y, z$  de cada uno de los vértices que forman dicha superficie.



Figura 4.1: Pantalla de bienvenida de la interfaz gráfica de usuario.

Archivos	Opciones de simulación
Su recinto se encuentra definido en la siguiente ubicación: Seleccione el archivo .ENV o .CAD <input type="text"/> <input type="button" value="Examinar"/>	Modelo <input checked="" type="radio"/> Cerrado <input type="radio"/> Interior <input type="radio"/> Abierto <input type="radio"/> Exterior <input type="button" value="Ayuda"/> Cantidad de texto a mostrar: 4
Los archivos generados en la simulación comenzarán por la palabra: <input type="text" value="20160916_090151"/>	Condición de finalización: Orden máximo de reflexión especular: <input type="text" value="2"/> Orden máximo de difracción en bordes: <input type="text" value="1"/>
Los archivos generados en la simulación se guardarán en: <input type="text"/> <input type="button" value="Examinar"/>	Respuesta al impulso <input type="checkbox"/> Calcular la respuesta al impulso Rstart [m]: <input type="text" value="1.3"/> Frecuencia de muestreo: <input type="text" value="96000"/> Velocidad del sonido [m/s]: <input type="text" value="344"/> Densidad del aire [kg/m3]: <input type="text" value="1.21"/>
Fuentes y receptores Introduzca el número de fuentes sonoras existentes: <input type="text" value="1"/> Introduzca las coordenadas de la ubicación de las fuentes [metros]: <input type="text"/> Formato: X1 Y1 Z1;X2 Y2 Z2;... Introduzca el número de receptores existentes: <input type="text" value="1"/> Introduzca las coordenadas de la ubicación de los receptores [metros]: <input type="text"/> Formato: X1 Y1 Z1;X2 Y2 Z2;...	<input type="button" value="Visualizar"/> <input type="button" value="Simular"/>

Figura 4.2: Pantalla de introducción de parámetros de la interfaz gráfica de usuario.

Por su parte, el formato `.cad` requerido por [EDT](#) y que se define a través de la herramienta de tipo comercial CATT-Acoustics, realiza una definición diferente de la geometría, es decir, en primer lugar define todos los vértices con sus respectivas coordenadas que componen la geometría, y en segundo lugar define cada una de las superficies en función de los vértices que ha definido previamente. No solo la forma de la definición influye para el correcto funcionamiento, sino que es necesario prestar una atención especial al sentido en el que se definen los vértices de cada plano ya que la normal de cada uno debe tener un signo específico como se verá a continuación.

La conversión de un formato a otro se realiza a través de dos funciones que son llamadas desde la interfaz de introducción de parámetros y son: `env2matlab` y `matlab2cad`.

**4.3.1.1.1 `env2matlab(ruta,envfile)`:** esta función se encarga de leer el archivo `.env` y los distintos archivos `.srf` asociados, creando un archivo denominado `data_struct.mat` donde se almacena la información que contienen los primeros. De esta estructura se destaca la variable `numsurfaces` que contiene el número de superficies de la geometría, y la estructura `surfaces` que contiene toda la información de vértices y planos.

**4.3.1.1.2 `matlab2cad()`:** esta función es clave para el correcto funcionamiento de la herramienta [EDT](#) ya que contiene los aspectos más importantes a tener en cuenta. Haciendo uso del archivo `data_struct.mat` generado por la función `env2matlab`, realiza varias tareas que se enumeran a continuación:

1. En primer lugar realiza la lectura de la estructura `surfaces` almacenando los datos en una serie de matrices ya con la forma necesaria para la herramienta [EDT](#). Destacan las siguientes matrices:
  - `cornernumbers` [número de vértices,1]: se trata de un vector columna en el que cada celda dispone un número consecutivo, según el número de vértices existentes.
  - `corners` [número de vértices,3]: se trata de una matriz en la que cada fila representa un vértice y en cada columna se da la coordenada de este vértice siguiendo el orden de definición típico  $x, y, z$ .
  - `planecorners` [número de superficies,número de vertices]: se trata de una matriz en el que cada fila representa un plano del recinto u obstáculo y se especifica el número de vértices por los que está compuesto cada plano.

Aunque en cada archivo `.srf` los vértices estén definidos en el orden correcto para que la normal de la superficie tenga el sentido deseado, al realizar esta lectura y tener que manipular los datos, es probable que no conserve el orden de definición, por tanto, se hacen necesarias las tres etapas siguientes.

2. Una vez organizados los datos según precisa el formato `.cad`, se comprueba que la matriz `planecorners` tenga los vértices de cada plano definidos en orden consecutivo, es decir, en sentido horario o antihorario. Para ello se utiliza el producto vectorial cruzado utilizando la función de [EDT](#) `EDB2cross`. Si el vector normalizado obtenido en el cálculo tiene todos valores a 1 o todos los valores a -1, la definición de los vértices es consecutiva y no habría que realizar ningún cambio. En caso que el vector contenga una mezcla de 1s y -1s, habrá que realizar una permutación de dos de los vértices del plano.
3. En tercer lugar se calcula el número de obstáculos que contiene la geometría incluyendo al recinto como un obstáculo más. Esto es necesario ya que la herramienta [EDT](#) tiene en cuenta el sentido de

las normales de cada superficie y en caso de que no se definan del modo que las requiere no realizará la simulación.

4. Una vez se conocen las superficies que forman el recinto y las que forman cada obstáculo, se vuelve a calcular la normal de cada superficie conociendo ya si pertenece al recinto o al obstáculo. En el caso del recinto, las normales deben apuntar hacia dentro si se quiere comprobar la acústica dentro del recinto, y en el caso de los obstáculos, el sentido de las normales de sus superficies deben estar apuntando hacia el exterior de los mismos, es decir, hacia las superficies del recinto. En la Figura 4.3 se comprueba un ejemplo de esto, se observa un recinto con geometría no paralelepípeda y en su interior un mueble y una columna. El mueble y la columna se consideran obstáculos y por tanto sus normales apuntan hacia fuera, en el caso del recinto se comprueba como las normales de los planos que lo forman disponen de normales apuntando hacia el interior del mismo.

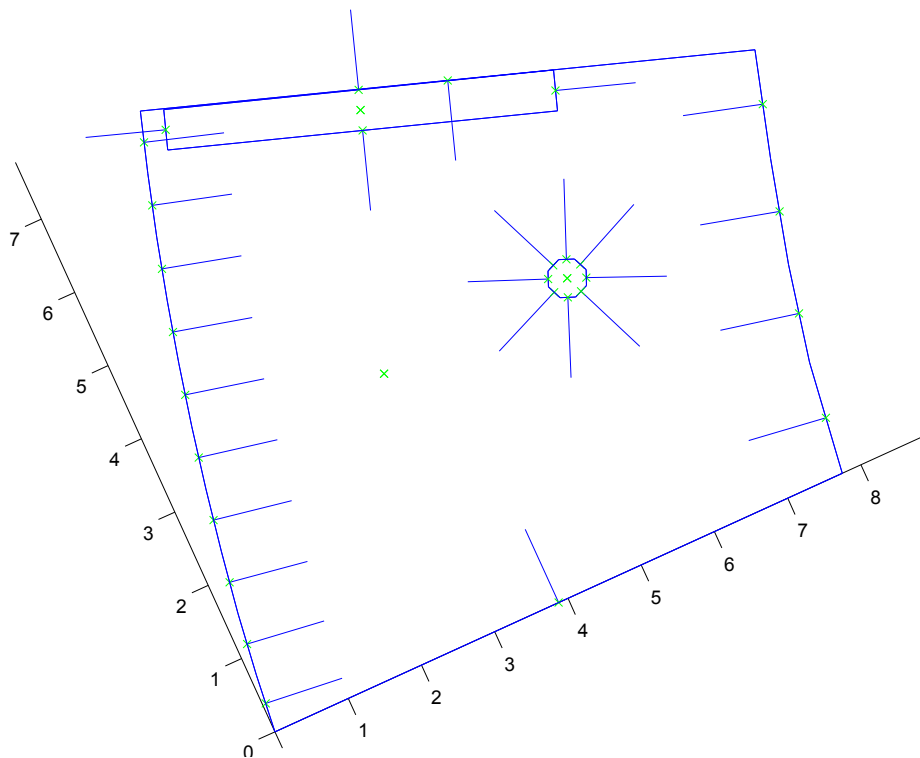


Figura 4.3: Ejemplo de sentido de las normales en un recinto con obstáculos

5. Una vez realizados los anteriores pasos con éxito, se crea un archivo denominado `environment.cad` con la información del recinto según la estructura de este tipo de formato. Se destaca que no se incluye la posición de las fuentes y receptores en este aunque es aceptado, ya que se ha incluido esta información en el archivo de configuración generado.

Estas dos funciones únicamente se utilizarán en el caso de que se introduzca un archivo con la geometría en formato `.env`. En caso de introducir un archivo `.cad` creado con CATT-Acoustics [44] no serán requeridas.

Se hace necesario destacar en este apartado la imposibilidad de generalizar el algoritmo de los puntos 2 a 4 en recintos como el mostrado en la Figura 4.3, es decir, con forma no paralelepípeda, siendo planteado este aspecto para trabajos futuros. En estos casos es necesario comprobar que el archivo `environment.cad` generado disponga de la definición correcta de los vértices de cada plano para que la normal apunte en el sentido deseado.

### 4.3.1.2 Creación del archivo de configuración

Los parámetros de simulación que deben encontrarse en `EDsetupfile` según se vio en el esquema 3.1, son obtenidos a través de las distintas opciones a seleccionar en el interfaz de usuario de la Figura 4.2. Éstos, junto a otros que no se da opción a seleccionar pero se encuentran definidos en el código de `simulador2`, se almacenan en un archivo denominado `info.mat`. Los parámetros en los que ha sido posible se ha seleccionado un valor por defecto con el objetivo de aligerar la ejecución de la herramienta.

La información relativa a las fuentes y receptores se almacena en un archivo denominado `sources_receivers.mat`.

### 4.3.1.3 Representación del modelo

En la Figura 4.2 se puede observar un botón denominado 'Visualizar' que puede ser usado al introducir la información del bloque Archivos, en cuyo caso se mostrará una nueva ventana con la representación del recinto y los obstáculos; y una vez se haya introducido la información relativa a las fuentes y receptores se representará también la posición de estos puntos. Para realizar esta representación se han definido las funciones `representation_room` y `representation_src_rcv`. La primera de ellas hace uso del archivo `data_struct.mat` generado en la función `env2matlab` y del archivo `info.mat` que contiene información de los parámetros introducidos en la interfaz. La segunda función por su parte recibe los valores necesarios por parámetros en la función.

## 4.3.2 Interfaz de programación

En la API desarrollada se ha creado una función denominada `InitSimulation` a la que se pasan los parámetros de simulación y así se crea una estructura de nombre `infos` con toda la información necesaria. Además, también se emplean las funciones `env2matlab` y `matlab2cad` especificadas en la subsección 4.3.1.1 en el caso de introducir un archivo con extensión `.env`.

## 4.4 Simulación

Por etapa de simulación se entiende la llamada a la función `EDB2main` de la herramienta EDT una vez se han preparado los datos necesarios. Se ha tratado de realizar las mínimas modificaciones posibles en el algoritmo de simulación de la herramienta, tratando en la etapa de preprocesamiento cumplir todos los requisitos impuestos por la misma. No obstante se destacan en la siguiente enumeración, las modificaciones realizadas:

- La interfaz gráfica y la de programación disponen de un acceso diferente a los parámetros de configuración, en el primer caso se realiza con una lectura a memoria de archivos `.mat` mientras que en la segunda no se desea lectura de memoria y la información es pasada a través de la estructura `infos` creada a tal efecto. Por tanto, se ha añadido un flag (`api`) con valor 0 si se ejecuta desde la interfaz gráfica y con valor 1 si se ejecuta desde la interfaz de programación. De este modo se realiza la llamada a la misma función `EDB2main`, cargando en un caso los datos desde los archivos `.mat` generados, y en otro caso desde la estructura realizada.
- La función `EDB2readcad` ha sido modificada del mismo modo que `EDB2main` ya que se diferencia la forma en la que se cargan los datos en función de si nos encontramos ante la interfaz gráfica o de programación.



- La herramienta [EDT](#) según se define en su manual, realiza el cálculo de la respuesta al impulso y de la función de transferencia, para ello dispone de dos variables (`calcirs` y `calctfs`) para seleccionar si se desea realizar estos cálculos o no.

En el sistema de simulación desarrollado en este [TFM](#) se ha implementado el cálculo de las respuestas al impulso, dando la opción en la interfaz de la Figura 4.2 de seleccionar si se desea o no su cálculo. Además, es posible introducir una serie de parámetros relativos a este cálculo. Cabe destacar la opción de la herramienta [EDT](#) de calcular las respuestas al impulso por varios métodos (Svensson, Vanderkooy o aproximación de Kirchoff), utilizando los dos últimos una función denominada `EDB2wedgelstcombo` la cual no viene incluida en los archivos de la herramienta. Por tanto, el cálculo de las [IR](#) se realiza por el primero de los métodos especificados.

Por su parte, el cálculo de la función de transferencia no ha sido incluido en el sistema de simulación resultante incluyendo en el archivo `info.mat` su variable con valor 0.

- Originalmente [EDT](#) contiene en su función principal `EDB2main` el cálculo del tiempo empleado en la simulación de cada uno de los bloques. Éstos han sido eliminados ya que en este sistema de simulación no aportan ninguna información relevante.

#### 4.4.1 Interfaz gráfica

Para realizar la simulación desde la interfaz gráfica existe un botón 'Simular' que llama al programa `EDB2main` y además almacena la información que arroja [EDT](#) en un archivo de texto. Cuando la simulación concluye se abre una nueva interfaz de visualización (véase Figura 4.4) donde desde el botón 'Resultados simulación' se podrá comprobar este archivo de texto.

#### 4.4.2 Interfaz de programación

En esta interfaz para realizar la simulación propiamente dicha, es decir, llamar a la función `EDB2main` se utiliza la función denominada `getpaths` a la que se le pasa por parámetro una matriz con las posiciones de la fuente o fuentes y otra con las del receptor o receptores. Además, también es necesario incluir la estructura `infos` creada en la etapa anterior.

### 4.5 Postprocesamiento

En esta sección se explicarán los desarrollos realizados tras la obtención de los resultados de la simulación, en el caso de la interfaz gráfica se mostrará la visualización de las trayectorias seguidas por los rayos en el recinto y en el caso de la interfaz de programación se almacenarán los puntos de esas trayectorias.

Cabe destacar que los resultados con interés en el sistema de simulación implementado se encuentran en los archivos `eddatafile` y `reflpathsfile` obtenidos tras la ejecución de las funciones pertinentes según se observa en el esquema de la Figura 3.1. Las trayectorias calculadas se encuentran en el segundo de los archivos especificados, destacando las matrices siguientes:

- `pathtypevec`: se trata de una matriz de tamaño [número de combinaciones,orden] donde se describe el tipo de fenómeno producido en cada orden, señalando el sonido directo con la letra 'f' o número 102, las reflexiones especulares con 's' o 115 y las difracciones en bordes con 'd' o 100.
- `reflpaths`: es una matriz que contiene el número de superficie o de borde que interviene en cada fenómeno especificado en `pathtypevec`.

- `specextradata`: contiene las coordenadas de las fuentes imagen (columnas 1-3), los receptores imagen (columnas 4-6) y las coordenadas de los puntos de intersección de los rayos con las superficies, estas últimas únicamente en el caso de trayectorias con solo reflexiones especulares.
- `firstdiffrow`: indica el número de fila dentro de las matrices `mainlistguide` y `mainlistguidepattern` donde existe la primera componente de difracción.
- `mainlistguide`: cada fila describe un tipo de combinación de trayectoria, especificando el número de filas de `reflpaths` y `pathtypevec` donde se tiene esa combinación.
- `mainlistguidepattern`: aporta una descripción de cada posible combinación definida en `mainlistguide` aportando la secuencia de reflexiones y/o difracciones.

### 4.5.1 Interfaz gráfica

Una vez concluida la simulación, se abre una nueva interfaz (`simulador3`) como la que se puede observar en la Figura 4.4. En esta es posible seleccionar la fuente y receptor que se desea representar en el caso de haber introducido más de uno ya que `EDT` almacena los resultados de simulación para cada par fuente-receptor en un archivo `edpathsfile` diferente.

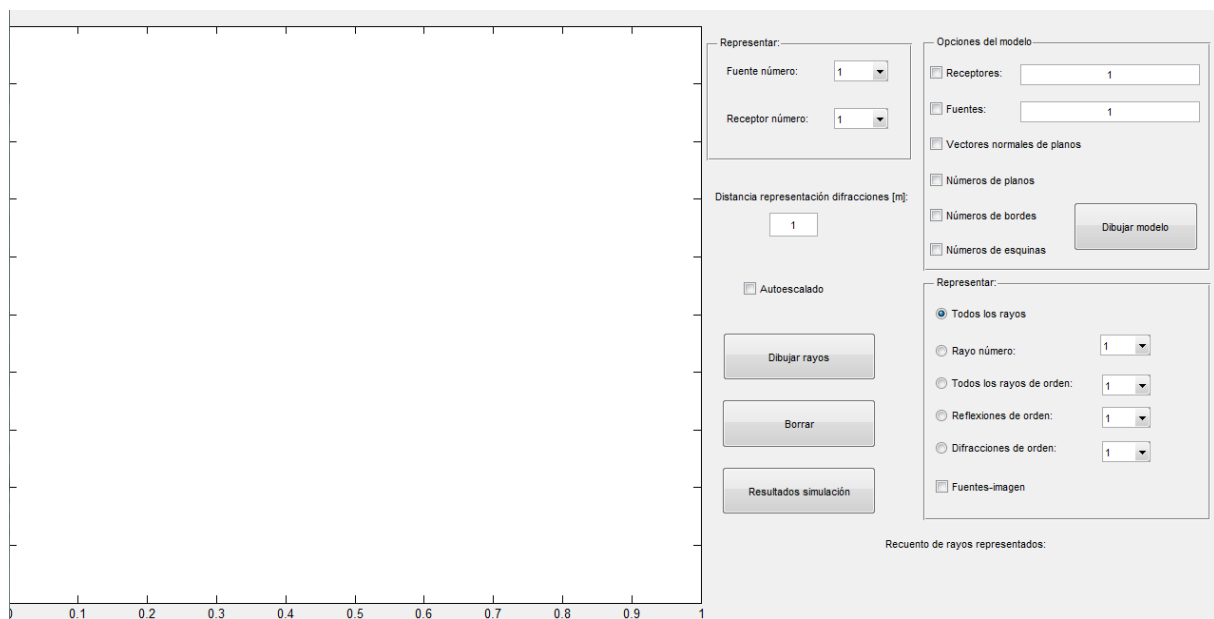


Figura 4.4: Pantalla visualización de resultados de la interfaz gráfica de usuario.

En el bloque 'Opciones de simulación' se puede seleccionar aquellas opciones que se desean representar al pulsar el botón 'Dibujar modelo'. Para representar estas opciones se hace uso de la función `EDB2plotmodel` perteneciente a la herramienta `EDT`.

En el bloque 'Representar' de la parte inferior derecha de la interfaz se puede seleccionar la opción de representación de rayos que se desee, la cual será representada en la parte izquierda de la interfaz al pulsar el botón 'Dibujar rayos'. Las funciones desarrolladas para la representación de cada una de estas opciones se especifican a continuación:

- La opción representar todos los rayos hace una llamada a la función con el siguiente prototipo:

```
[n_rays, raytype, paths]
```

```
= representation_paths(eddatafile, reflpathsfile,
                        fuente_imagen,
                        plot_source, plot_receive,
                        separacion_point_edges, api)
```

Esta función representa todos los rayos que han sido calculados en la simulación devolviendo el número de rayos total que ha sido representado, así como otros dos parámetros que se analizarán en la sección 4.5.2. Los parámetros de entrada tienen el siguiente significado:

- `fuente_imagen`: tendrá un valor 1 cuando se haya seleccionado en la interfaz que se desea representar las fuentes imagen y en caso contrario tendrá un valor 0.
- `separacion_point_edges`: indica la separación en metros a la que se quiere representar un rayo difractado dentro de un borde. Es decir, la herramienta [EDT](#) únicamente determina si la trayectoria de un rayo pasará por un determinado borde pero no aporta el punto exacto por donde pasará. Con el objetivo de representar este fenómeno de forma gráfica se escogen varios puntos del borde y se representan los rayos. Siempre se elige el punto medio del borde y a partir de él se suma a ambos lados el valor en metros de esta variable. Si el nuevo punto está fuera de la longitud del borde, es desechado, en caso contrario, se vuelve a sumar dicha distancia. A mayor valor de esta variable menos rayos se representarán por borde y a menor valor mayor número de ellos.
- `api`: se trata del flag comentado anteriormente que sirve para distinguir si se está ante la interfaz gráfica o la interfaz de programación. En el primer caso se realizará una llamada a `plot3` de Matlab y en el segundo se devolverán los parámetros `raytype` y `paths`.

- La opción de representar un número de rayo concreto hace una llamada a la siguiente función:

```
representation_onepath(eddatafile, reflpathsfile, fuente_imagen,
                        num_rayo, plot_source, plot_receive,
                        separacion_point_edges)
```

Esta función no devuelve ningún parámetro pero representa en la interfaz el rayo que ha sido seleccionado. Los parámetros que se le pasan a la función ya han sido definidos anteriormente.

- La opción de representar solo reflexiones de un orden determinado hace una llamada a la función siguiente:

```
rep=representation_specularpaths(eddatafile, reflpathsfile,
                                order, fuente_imagen)
```

Esta función representa únicamente los rayos especulares puros del orden deseado, es decir, en el momento que la trayectoria tenga una difracción no será representado en esta opción. Devuelve el vector `rep` con los números de rayos representados.

- La opción de representar difracciones de un orden determinado hace una llamada a la función:

[illegible]

Esta función representa los rayos del orden especificado que contienen al menos una difracción, es decir, ante una trayectoria con un orden especular y otro orden de difracción, se dibujará dicha trayectoria si se selecciona la opción de representar difracciones de orden 1. En el caso de seleccionar orden 2 se representarán las trayectorias que contengan dos órdenes de difracción. Devuelve el vector `dibujar` que contiene el número de rayos representados.

Como se ha adelantado, la herramienta EDT únicamente determina el borde en el que existe difracción pero no aporta puntos exactos para representar los rayos. Con el objetivo de que en la representación gráfica se vean reflejadas las difracciones en los bordes, se calculan una serie de puntos a lo largo del borde y además es necesario calcular el punto de intersección del rayo reflejado con la superficie. Este último aspecto tiene que ver con que la herramienta solo aporta la fuente imagen o receptor imagen cuando la trayectoria está formada por algún orden de difracción. Para calcular la intersección de una recta que une el punto del borde con la fuente o receptor imagen y un plano se hace uso de la función `intersectLinePlane` obtenida en el intercambiador de archivos de Mathworks [95]. Esta función realiza una llamada a la función `vectorCross3d` la cual se encuentra también disponible en [95].

- La opción de representar todos los rayos de un orden determinado hace una llamada a las funciones `representation_specularpaths` y `representation_diffpaths` explicadas en los puntos anteriores.

La opción de representar las fuentes imagen se tiene en cuenta en cada una de las funciones ya explicadas.

### 4.5.2 Interfaz de programación

Los resultados de la simulación en el caso de la interfaz de programación se desean en forma de matriz o estructura, de forma que no sea necesario el acceso a memoria para obtener estos datos. De este modo no se realiza un almacenamiento de los datos en un archivo `.mat`.

La generación de estos datos se realiza dentro de la función `representation_paths` explicada en el apartado 4.5.1. Esta función devuelve una matriz denominada `raytype` y otra de nombre `paths`. La primera contiene los tipos de fenómenos producidos en la trayectoria de un rayo, y la segunda contiene los puntos por los que discurre la trayectoria de cada rayo, para cada una de las fuentes y receptores especificados en la simulación. Hay que destacar que en estas variables se incluyen las trayectorias calculadas para cada difracción en borde, es decir, como se ha comentado, a través de la variable `separacion_point_edges` se determinan una serie de rayos y son estos los especificados en estas matrices. El formato de estos datos se explica más detalladamente en el Apéndice B.

Por tanto, la función `getpaths` perteneciente a la API devuelve las estructuras con las matrices calculadas en la función `representation_paths` para cada fuente y receptor especificado. Se incluye un script de Matlab de nombre `api.m` con un ejemplo de utilización de la interfaz de programación.

## 4.6 Aspectos a considerar en el sistema de simulación

Caben destacar algunos aspectos a tener en cuenta para el correcto funcionamiento del sistema, estos son:

- Las modificaciones en la geometría del recinto y obstáculos debe ser realizadas en el archivo `.env`, es decir, si están especificadas en él un número de superficies `.srf` y no se desea utilizar todas, se

indicará que el número de superficies será el número  $N$  deseado de éstas, de esta forma únicamente se leerán las  $N$  primeras.

- Las primeras superficies (`.srf`) definidas en el `.env` deben ser aquellas que forman la geometría del recinto, y a continuación deben ser definidas las superficies de los obstáculos. Además, se recomienda que no se intercalen superficies de distintos obstáculos.
- Los obstáculos tienen que tener volumen, no es suficiente un plano para representar un obstáculo ya que de ese modo no se pueden determinar los bordes.
- En el caso de disponer de un recinto con geometría no paralelepípeda en el que existan planos u obstáculos con más de cuatro vértices no se garantiza el correcto funcionamiento de la función `matlab2cad`. Por tanto se recomienda comprobar el archivo `environment.cad` generado por ésta, comprobando especialmente si el orden de definición de los vértices en cada plano es correcto para obtener la normal con el sentido especificado en el apartado 4.3.1.1. Una vez realizado esto se procedería a llamar al archivo `.cad` modificado.
- Las tres coordenadas  $x, y, z$  que definen la posición de fuentes y/o receptores no pueden tener un valor simultáneamente de 0, es decir, la fuente o receptor no puede estar situado en la posición 0,0,0, al menos una de las coordenadas tiene que tener un valor distinto, por ejemplo 0.001.
- Dependiendo de la ubicación de la fuente y receptor puede aparecer un error en la simulación que dice que no está implementada la comprobación de la obstrucción entre bordes alineados. Para solucionarlo se puede especificar un orden de difracción 0 o bien modificar la ubicación de la fuente y receptor.
- El orden de difracción debe ser menor o igual que el de reflexión. Para reflexiones de orden 3 o superior la difracción debe ser 0. Esto es debido a que al calcular difracciones de orden 3 o superiores se produce un error de programación en la herramienta EDT.
- Las modificaciones en la geometría del recinto y obstáculos debe ser realizadas en el archivo `.env`, es decir, si están especificadas en él un número de superficies `.srf` y no se desea utilizar todas, se indicará que el número de superficies será el número  $N$  deseado de éstas, de esta forma únicamente se leerán las  $N$  primeras.
- El primer elemento definido en el `.env` debe ser el recinto y a continuación todos los obstáculos. Además, se recomienda que no se intercalen archivos de distintos obstáculos.
- Los obstáculos tienen que ser definidos en 3D, no es suficiente un plano para representar un obstáculo ya que de ese modo no se pueden determinar los bordes.
- En el caso de disponer de un recinto con geometría no paralelepípeda en el que existan planos u obstáculos con más de cuatro vértices no se garantiza el correcto funcionamiento de la función `matlab2cad`. Por tanto se recomienda comprobar el archivo `environment.cad` generado por ésta, comprobando especialmente si el orden de definición de los vértices en cada plano es correcto para obtener la normal con el sentido especificado en el apartado 4.3.1.1. Una vez realizado esto se procedería a llamar al archivo `.cad` modificado.
- Las tres coordenadas  $x, y, z$  que definen la posición de fuentes y/o receptores no pueden tener un valor simultáneamente de 0, es decir, la fuente o receptor no puede estar situado en la posición 0,0,0, al menos una de las coordenadas tiene que tener un valor distinto, por ejemplo 0.001.

- Dependiendo de la ubicación de la fuente y receptor puede aparecer un error en la simulación que dice que no está implementada la comprobación de la obstrucción entre bordes alineados. Para solucionarlo se puede especificar un orden de difracción 0 o bien modificar la ubicación de la fuente y receptor.
- El orden de difracción debe ser menor o igual que el de reflexión y para reflexiones de orden 3 o mayor la difracción debe ser 0.

## 4.7 Conclusiones

A la hora de realizar el sistema de simulación acústico a partir de una herramienta ya creada como es [EDT](#), se han encontrado múltiples aspectos a solucionar, entre ellos y con una especial relevancia la cuestión del sentido de las normales de cada plano. Además, ha sido realizado un gran esfuerzo en comprender el funcionamiento de esa herramienta para poder ser integrada en el sistema de simulación desarrollado.

Como aspectos de relevancia tras la realización del sistema de simulación, cabe destacar que se han alcanzado los siguientes objetivos:

- Se ha creado una interfaz gráfica de usuario, así como una interfaz de programación de aplicaciones, ambas utilizando el sistema de simulación implementado.
- Se ha incluido en el sistema la opción de introducir geometrías definidas en `.env/.srf` pero únicamente para recintos paralelepípedos.
- El sistema de simulación es capaz de incluir recintos con geometría arbitraria y múltiples obstáculos.
- El sistema de simulación permite la introducción de las posiciones de las fuentes y receptores, así como introducir más de una de ellas en cada simulación.
- También permite seleccionar algunos parámetros de simulación, como es la condición de finalización del algoritmo basada en el orden de las trayectorias o el cálculo o no de la respuesta al impulso.
- Se han obtenido las trayectorias de los rayos acústicos tanto de forma gráfica como en forma de datos para poder ser utilizada esta información en otros proyectos.

## Capítulo 5

# Evaluación y resultados

### 5.1 Introducción

En este capítulo se mostrarán algunos ejemplos de los resultados obtenidos a través de la ejecución del sistema de simulación desarrollado, mostrando principalmente los resultados en la interfaz gráfica por ser más visuales que en la interfaz de programación. Se ha evaluado el sistema de simulación con tres recintos cerrados diferentes, por tanto en todos ellos se ha seleccionado un modelo cerrado e interior.

El capítulo se ha estructurado en tres secciones, cada una de ellas dedicada a uno de los recintos especificados. El primero de ellos se trata de un recinto sencillo con geometría paralelepípeda compuesto únicamente por una mesa en su interior. El segundo de los recintos se trata de una sala con una pared en su interior que divide la habitación en dos zonas. El tercero de los recintos se trata de una geometría no paralelepípeda, compuesta por dos obstáculos como son un mueble pegado a la pared y una columna representada por un octógono. En cada una de las secciones se comprobarán los resultados de la simulación para varias ubicaciones de fuentes y receptores, además de mostrar varias opciones en los órdenes de reflexión y difracción.

### 5.2 IdiapRoom

El entorno IdiapRoom (véase Figura 5.1) es un recinto cerrado de 8,2 metros de largo, 3,6 metros de ancho y 2,4 metros de alto. Dispone en su interior de una mesa de 4,8 metros de largo y 1,2 metros de ancho. Originalmente, la mesa no dispone de espesor, pero como se ha contemplado en el apartado 4.6 es necesario que disponga de él, por tanto se ha representado con un espesor de 7 centímetros.

#### 5.2.1 Caso 1

En primer lugar se comprobarán los resultados obtenidos tras la simulación del recinto IdiapRoom colocando tanto la fuente como el receptor encima de la mesa, disponiendo entre ellos de visión directa. Según se observa en la Figura 5.2, la fuente (circunferencia roja) se sitúa a la derecha del receptor (circunferencia verde) y ambos se encuentran alineados. En la Tabla 5.1 se pueden comprobar las coordenadas de ambos puntos.

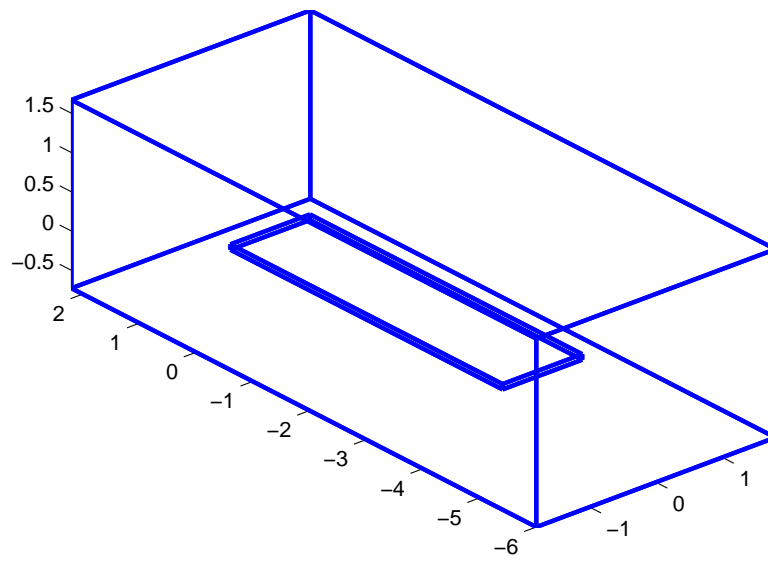


Figura 5.1: Recinto IdiaproRoom

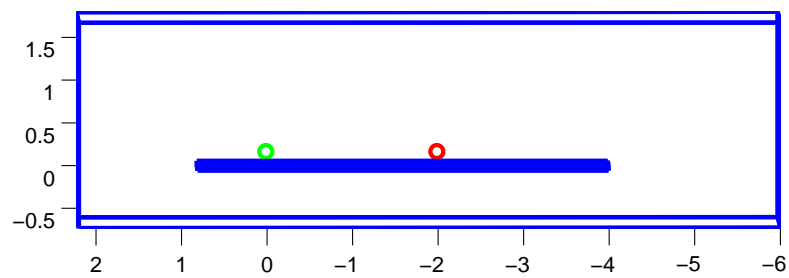


Figura 5.2: Posición fuente y receptor caso 1 IdiaproRoom

	Coordenada $x$	Coordenada $y$	Coordenada $z$
Fuente	0	-2	0.1
Receptor	0	0	0.1

Tabla 5.1: Posición fuente y receptor caso 1 IdiaproRoom



### 5.2.1.1 Reflexión y difracción de orden 2

En la Figura 5.3 se observa la representación de los 79 trayectos calculados por la herramienta de simulación para un orden 2 tanto de reflexión como de difracción. Además se ha seleccionado una separación entre rayos en cada borde de 1 metro.

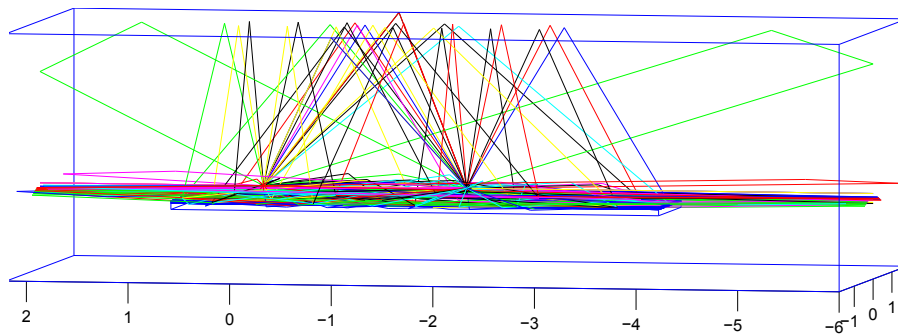


Figura 5.3: Simulación IdiapRoom con orden 2 de reflexión y orden 1 de difracción

En la Figura 5.4 se pueden observar los tres tipos de trayectos con difracción que pueden darse en este recinto. A la izquierda se encuentra representada la fuente en asterisco negro, la cual emite un rayo que refleja en la pared derecha e incide en el borde de la mesa opuesto, llegando así al receptor con orden 2. En el caso central se puede comprobar lo contrario, desde la fuente el rayo incide en un borde y realiza una reflexión en la pared derecha para así llegar al receptor. En la imagen derecha se puede comprobar un orden 2 de difracción ya que la trayectoria llega a dos bordes diferentes antes de llegar al receptor.

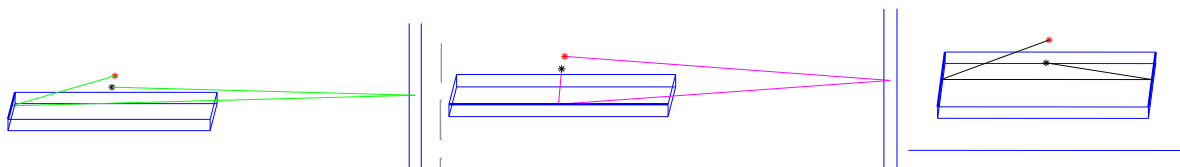


Figura 5.4: Tipos de difracciones en IdiapRoom

### 5.2.1.2 Reflexión de orden 6

Por su parte, en la Figura 5.5 se puede comprobar como seleccionando un orden 6 de reflexión sin cálculo de difracción, existen 298 trayectorias diferentes, entre las que se encuentran rayos de orden 1 hasta orden 6 de reflexión especular.

## 5.2.2 Caso 2

En este caso se sitúa el receptor debajo de la mesa según se puede observar en la Figura 5.6 en color verde. En la Tabla 5.2 se pueden comprobar las coordenadas de ambos.

### 5.2.2.1 Reflexión y difracción de orden 2

Realizando una simulación de orden 2 tanto en reflexión como en difracción, se pueden comprobar 22 trayectorias calculadas en la Figura 5.7, con 3 de ellas de orden 1 de reflexión y 7 de orden 2. Se ha

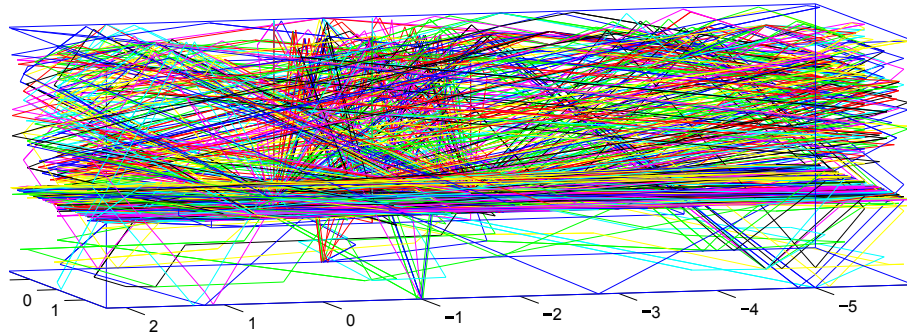


Figura 5.5: Simulación IdiapRoom con orden 6 de reflexión

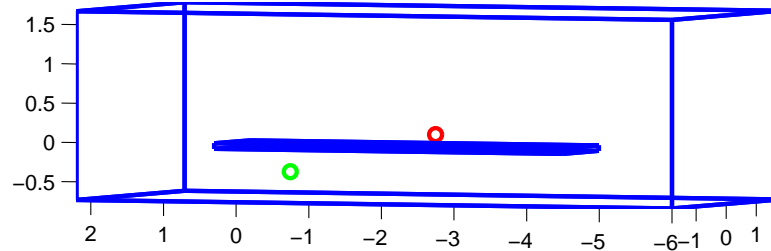


Figura 5.6: Posición fuente y receptor caso 2 IdiapRoom

	Coordenada $x$	Coordenada $y$	Coordenada $z$
Fuente	0	-2	0.1
Receptor	0	0	-0.4

Tabla 5.2: Posición fuente y receptor caso 2 IdiapRoom

seleccionado 1 metro de distancia entre puntos en cada borde para representar los rayos.

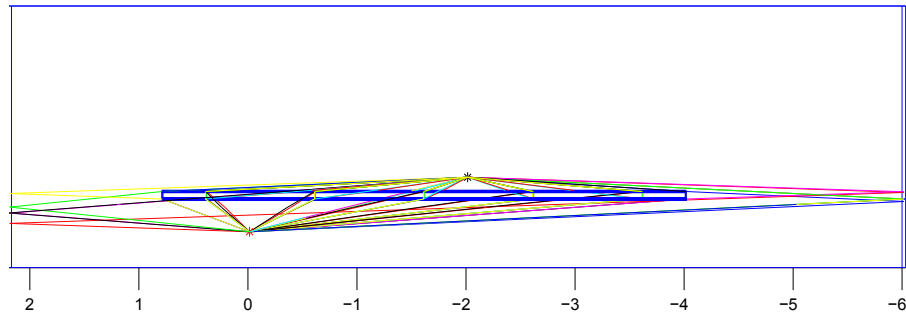


Figura 5.7: Simulación IdiapRoom caso 2 con orden 2 de reflexión y orden 2 de difracción

#### 5.2.2.2 Reflexión de orden 8

Realizando una simulación con orden 8 de reflexión y sin difracción, se muestra en la Figura 5.8 los 555 trayectos calculados desde una vista aérea del recinto. Hay que destacar que este es el máximo orden que se puede calcular con el ordenador en el que se ha realizado este TFM sin que se obtenga un error de memoria.

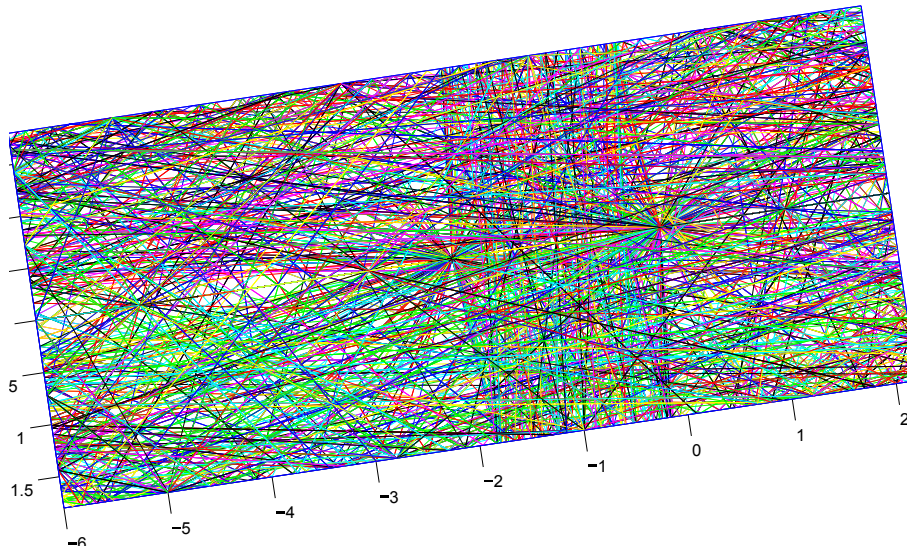


Figura 5.8: Simulación IdiapRoom caso 2 con orden 8 de reflexión

### 5.3 demoRoom3

El recinto demoRoom3 (véase Figura 5.9) se trata de un recinto cerrado de 6,2x6,2x5 metros que contiene en su interior una pared de 1 metro de espesor que separa la sala en dos zonas.

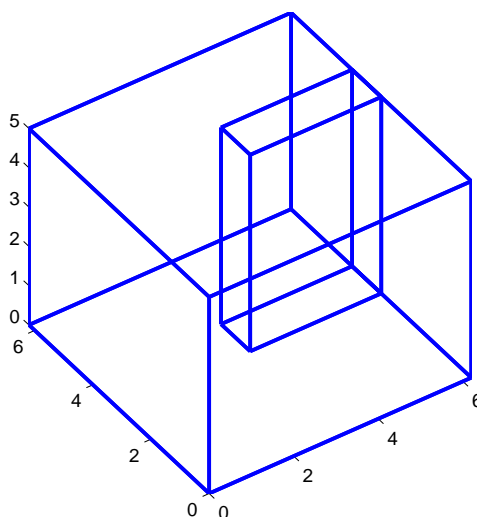


Figura 5.9: Recinto demoRoom3

### 5.3.1 Caso 1

En este caso se sitúa la fuente y receptor cada uno a un lado de dicha pared, según se observa en la Figura 5.10 y se muestran las coordenadas en la Tabla 5.3. El receptor está representado en color verde y la fuente en color rojo.

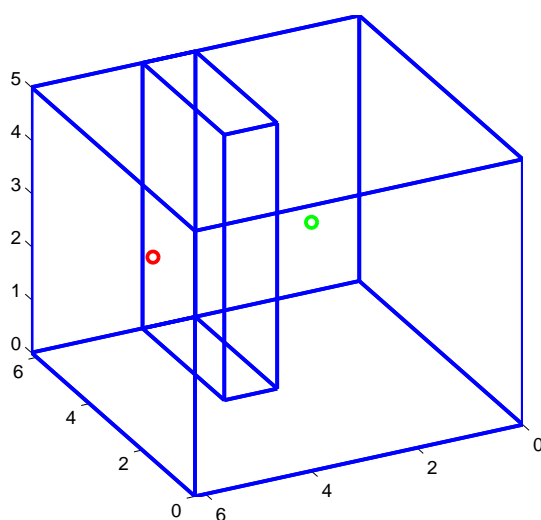


Figura 5.10: Posición fuente y receptor demoRoom3

	Coordenada $x$	Coordenada $y$	Coordenada $z$
Fuente	4	5	2.5
Receptor	4	2	2.5

Tabla 5.3: Posición fuente y receptor demoRoom3

### 5.3.1.1 Reflexión y difracción de orden 2

En la Figura 5.11 se pueden comprobar 9 trayectorias diferentes teniendo en cuenta que aquellas en las que existe algún orden de difracción, se ha seleccionado una separación entre puntos del borde de 1 metro para la representación.

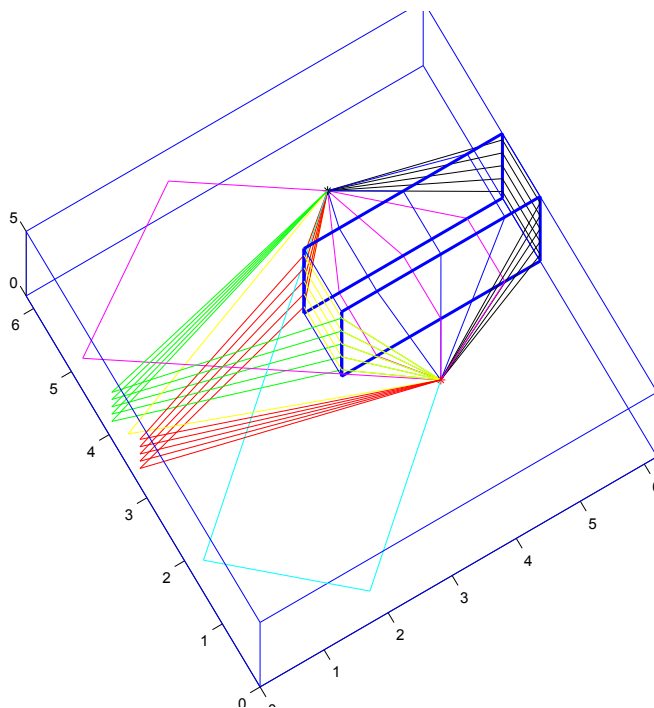


Figura 5.11: Simulación demoRoom3 con orden 2 de reflexión y difracción

### 5.3.1.2 Reflexión de orden 9

En la Figura 5.12 se comprueban los 424 trayectos calculados por la herramienta de simulación para un orden 9 de reflexión, siendo este orden el mayor que puede ser simulado con el ordenador en este recinto.

## 5.4 Ispace

El recinto Ispace (véase Figura 5.13) se trata de un recinto cerrado situado en la Escuela Politécnica Superior de la Universidad de Alcalá. Su geometría se asemeja a la sección de una circunferencia, disponiendo así de dos paredes curvas. Las pared curva más pequeña tiene una longitud de 5,23 metros, mientras que la opuesta tiene una longitud de 7,71 metros. Las caras laterales planas tienen una medida de 7,75 metros y de 7,27 metros. En el interior de este recinto se encuentra una columna representada en forma octogonal y con aproximadamente medio metro de diámetro. Además, también se encuentra en el interior un segundo obstáculo, más concretamente un armario pegado a la pared. Éste tiene una anchura aproximada de medio metro, no llegando su altura al techo de la pared.

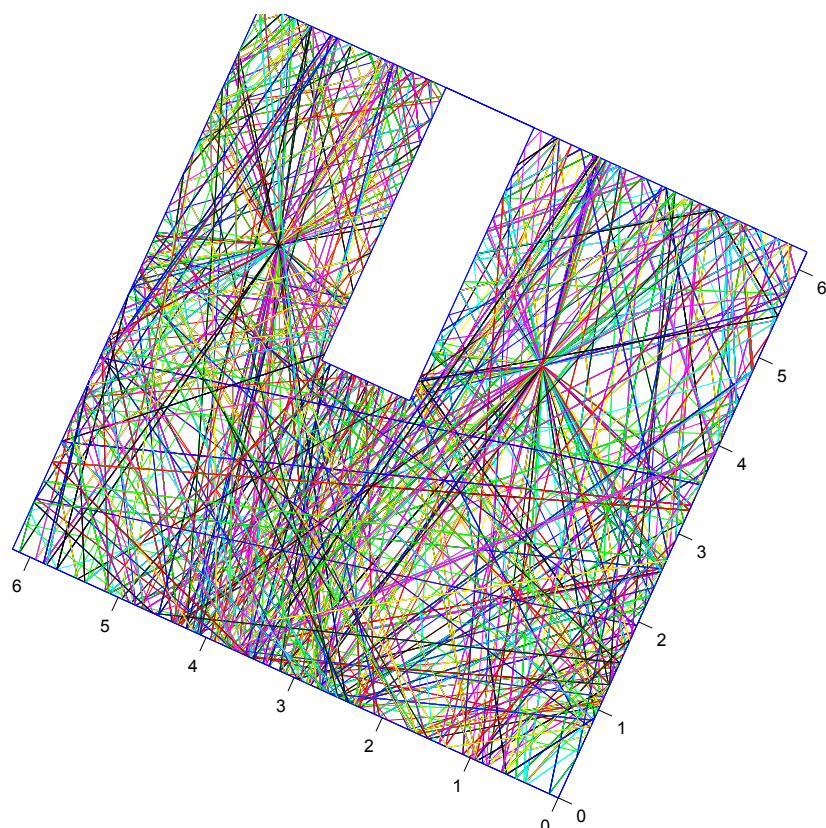


Figura 5.12: Simulación demoRoom3 con orden 9 de reflexión

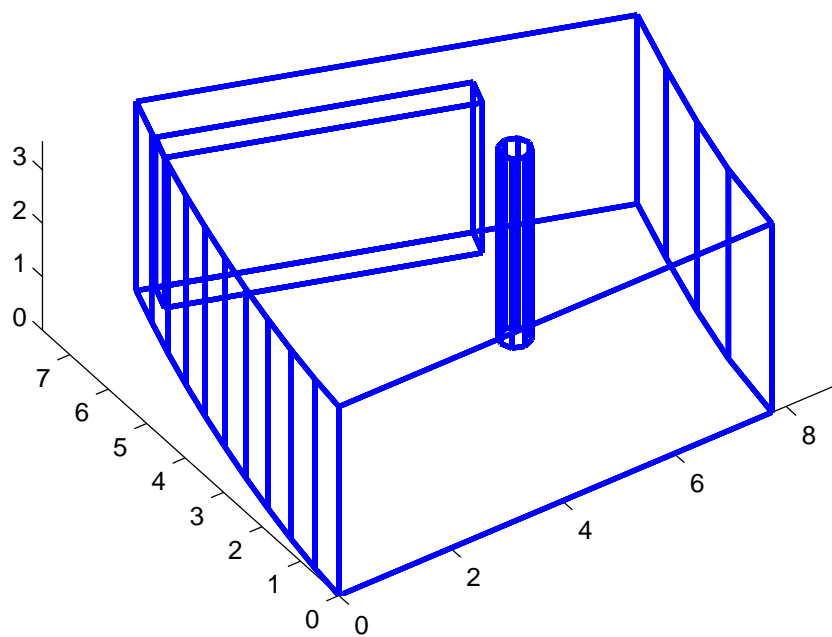


Figura 5.13: Recinto Ispace

### 5.4.1 Caso 1

En este caso se sitúa al receptor cerca de la columna (círculo color verde) y a la fuente en el otro extremo, según se observa en la Figura 5.14. Por su parte, en la Tabla 5.4 se especifican las coordenadas tanto de la fuente como del receptor.

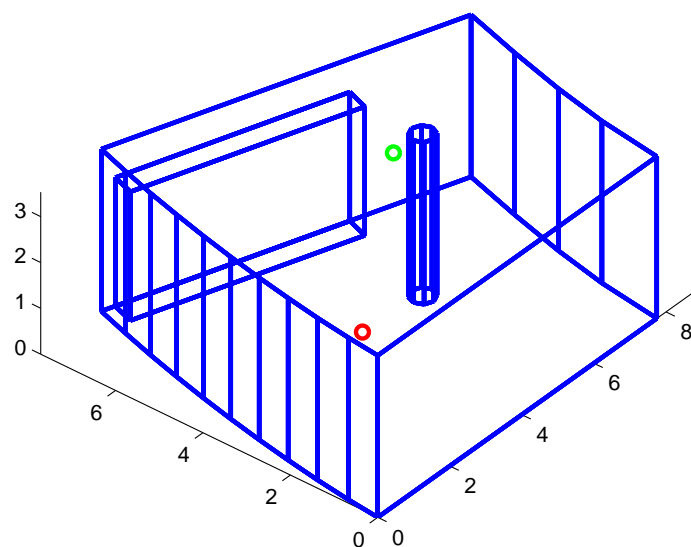


Figura 5.14: Posición fuente y receptor caso 1 Ispace

	Coordenada $x$	Coordenada $y$	Coordenada $z$
Fuente	2	2	2
Receptor	6.5	5	2

Tabla 5.4: Posición fuente y receptor caso 1 Ispace

#### 5.4.1.1 Reflexión y difracción de orden 2

En la Figura 5.15 se observan los 845 trayectos calculados por la herramienta de simulación, seleccionando una distancia entre puntos de cada borde de 1 metro. Cabe destacar el hecho de que por el centro de la columna existen trayectos, esto no debería ocurrir en la realidad al estar este obstáculo compuesto por superficies rígidas que reflejan toda la energía que reciben. Para explicar este hecho basta con observar la Figura 5.16 en la que se comprueba como la simulación determina que existe difracción en un borde inferior de la columna y en el borde opuesto de la misma en el mismo trayecto. Para solucionar este aspecto derivado bastaría con no simular difracciones en el interior del recinto.

#### 5.4.1.2 Reflexión de orden 5

En la Figura 5.17 se representan los 182 trayectos calculados con un orden 5 seleccionado de reflexión. Este es el máximo orden a simular sin que aparezca error de memoria



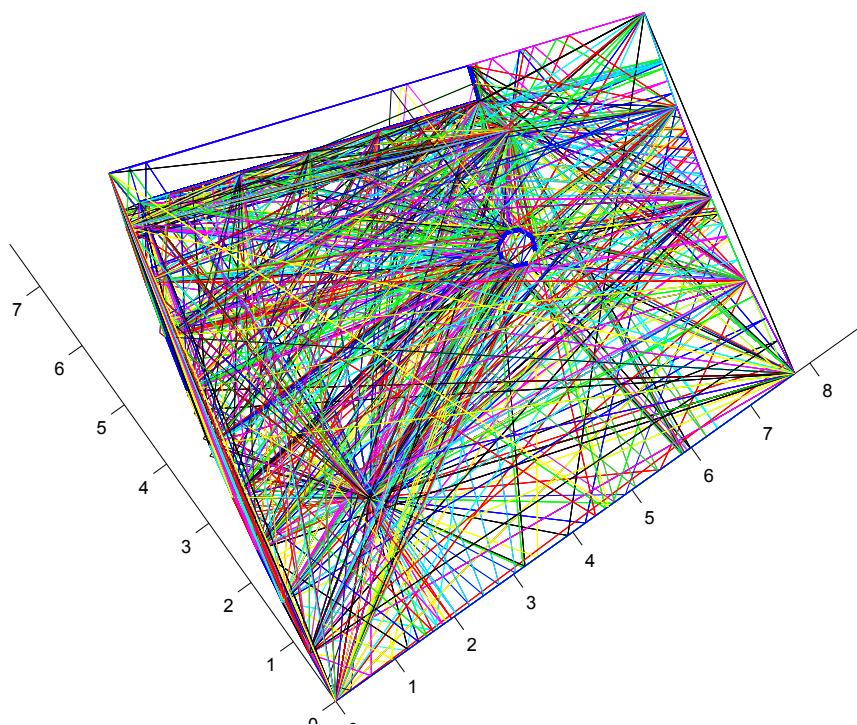


Figura 5.15: Simulación Ispace caso 1 con orden 2 de reflexión y difracción

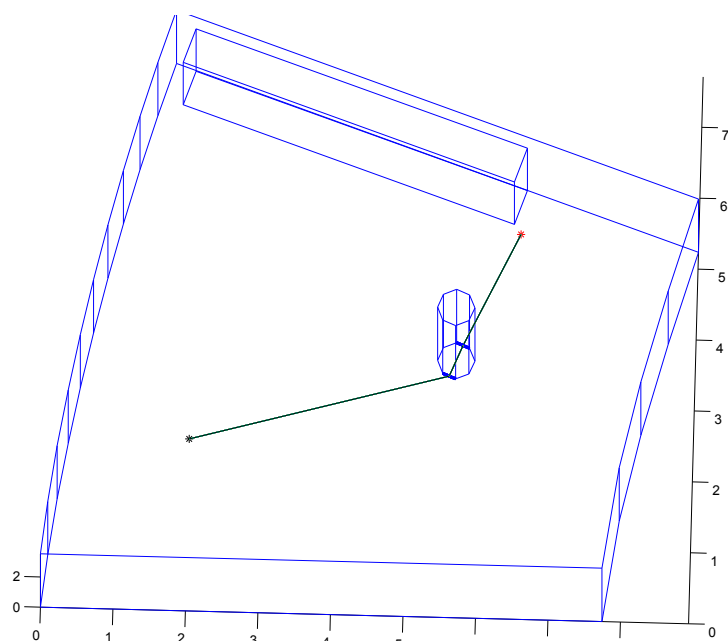


Figura 5.16: Detalle trayectoria Ispace con dos órdenes de difracción



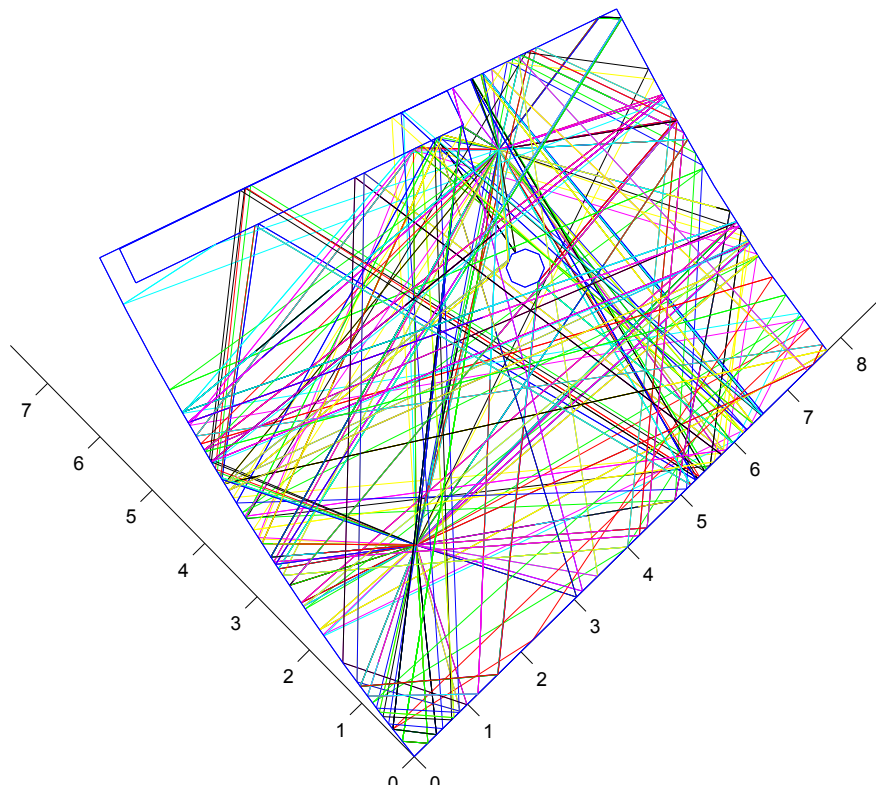


Figura 5.17: Simulación Ispace caso 1 con orden 5 de reflexión

## 5.4.2 Caso 2

En este caso se coloca el receptor en la posición donde está anclado un array de micrófonos y la fuente cerca del armario, según se observa en la Figura 5.18. En la Tabla 5.5 se muestran las coordenadas de estos puntos.

	Coordenada $x$	Coordenada $y$	Coordenada $z$
Fuente	2	6	1.6
Receptor	3.9	0.02	2.26

Tabla 5.5: Posición fuente y receptor caso 2 Ispace

### 5.4.2.1 Reflexión y difracción de orden 2

En la Figura 5.19 se pueden comprobar los 874 trayectos calculados por la herramienta de simulación con un orden de reflexión y difracción 2. Al igual que en los anteriores casos, se ha seleccionado una distancia entre puntos a representar del borde de 1 metro.

### 5.4.2.2 Reflexión de orden 5

Por su parte, en la Figura 5.20 se observan un total de 159 trayectorias calculadas.

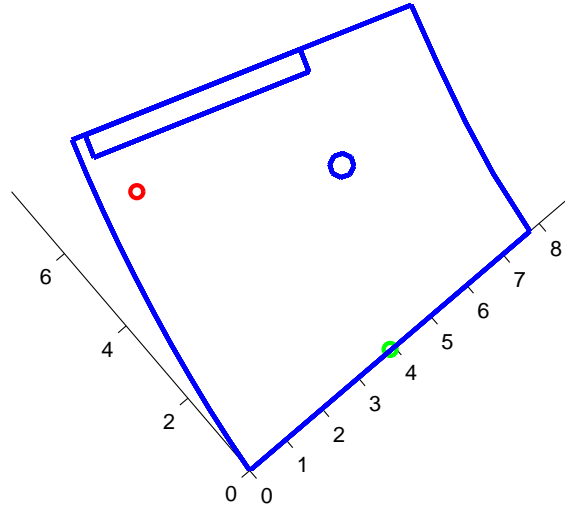


Figura 5.18: Posición fuente y receptor caso 2 Ispace

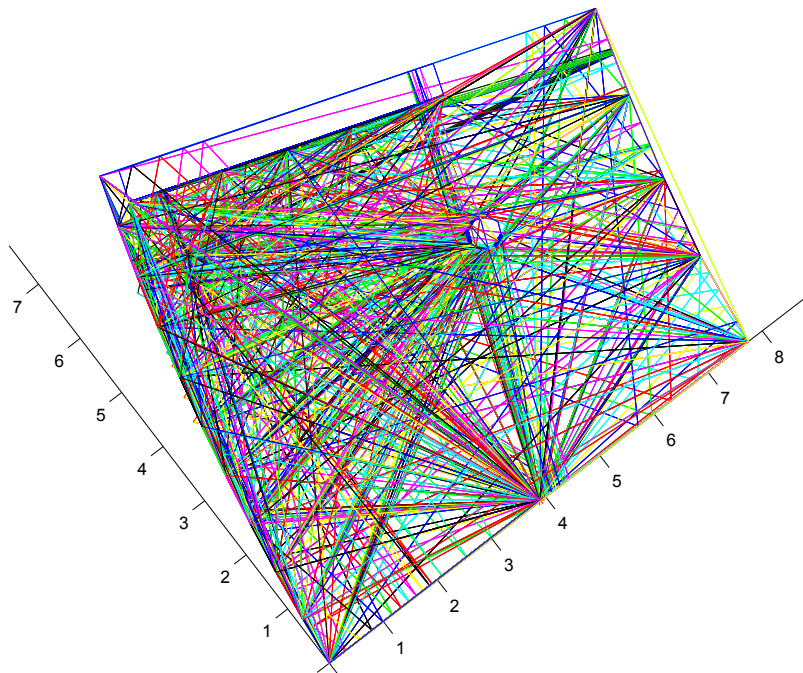


Figura 5.19: Simulación Ispace caso 2 con orden 2 de reflexión y difracción

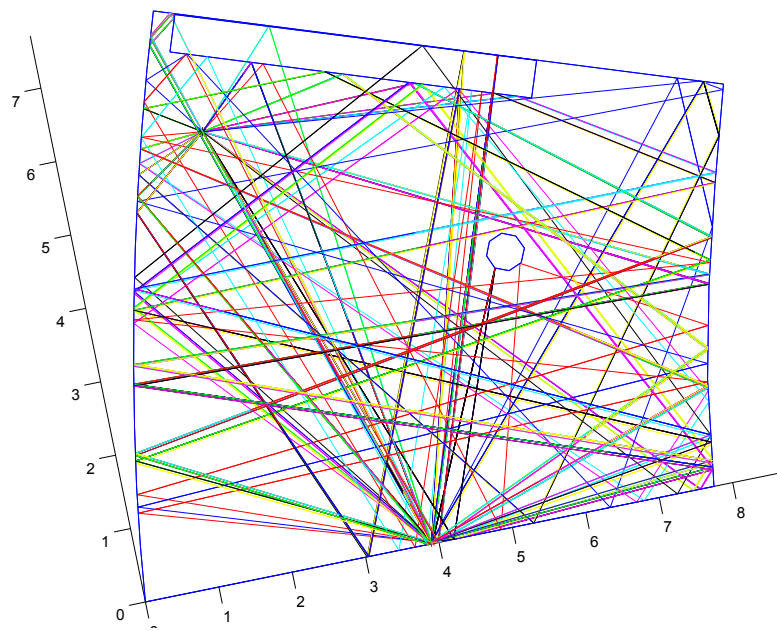


Figura 5.20: Simulación Ispace caso 2 con orden 5 de reflexión

### 5.4.3 Caso 3

En este último caso se sitúa el receptor (array de micrófonos) en una de las paredes curvas del recinto y la fuente se sitúa de nuevo en los alrededores del armario, según se puede comprobar en la Figura 5.21. En la Tabla 5.6 se especifican las coordenadas de la posición tanto de la fuente como el receptor.

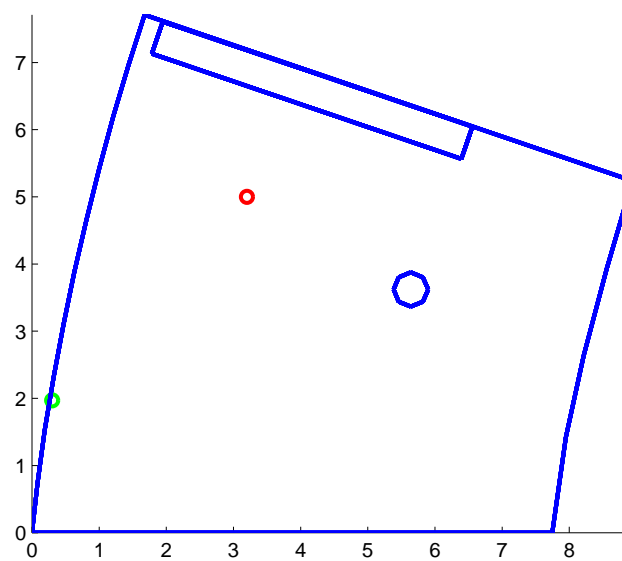


Figura 5.21: Posición fuente y receptor caso 3 Ispace

	Coordenada $x$	Coordenada $y$	Coordenada $z$
Fuente	3.2	5	1.6
Receptor	0.3	1.97	2.27

Tabla 5.6: Posición fuente y receptor caso 3 Ispace

#### 5.4.3.1 Reflexión y difracción de orden 2

En la Figura 5.22 se muestran los 998 trayectos calculados por la herramienta de simulación para un orden 2 de reflexión y de difracción.

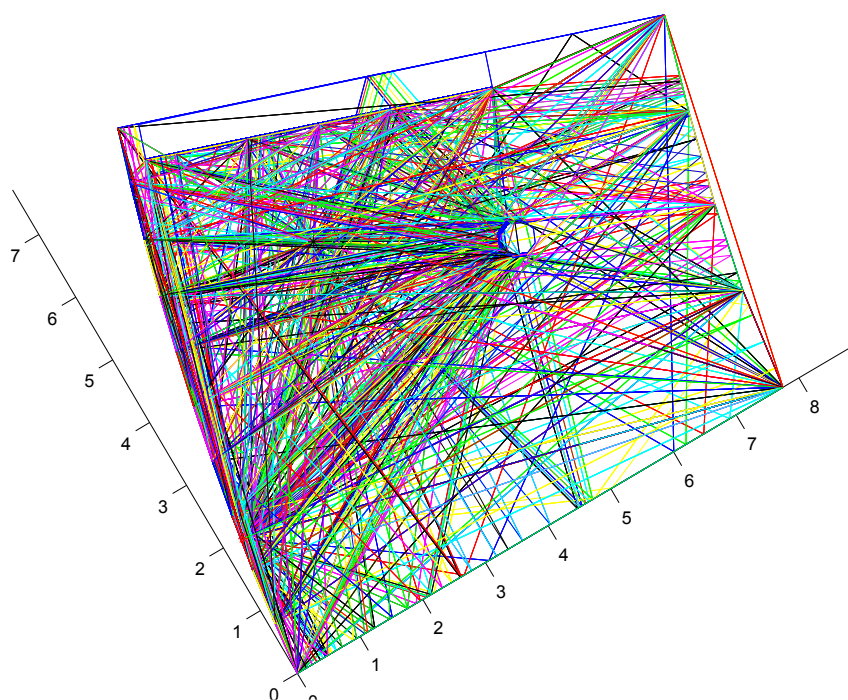


Figura 5.22: Simulación Ispace caso 3 con orden 2 de reflexión y difracción

#### 5.4.3.2 Orden 5 de reflexión

En último lugar, en la Figura 5.23 se pueden observar un total de 186 trayectos con un orden 5 de reflexión, el máximo posible en la simulación con este ordenador sin obtener error de memoria.

## 5.5 Conclusiones

Tras la realización de una serie de pruebas con el sistema de simulación implementado en este TFM, es posible establecer una serie de conclusiones acerca de su funcionamiento.

Uno de los principales aspectos a destacar y que supone una de las principales limitaciones del sistema es que no se puede seleccionar cualquier orden de reflexión ya que dependiendo de la geometría del recinto y del procesador donde se ejecute la simulación, se obtendrá un error de memoria a un orden determinado.

Por otra parte, se destaca el hecho de incluir obstáculos como es una columna representada a través de ocho planos. Al simular con difracciones en bordes, las uniones de esos planos que simulan la columna

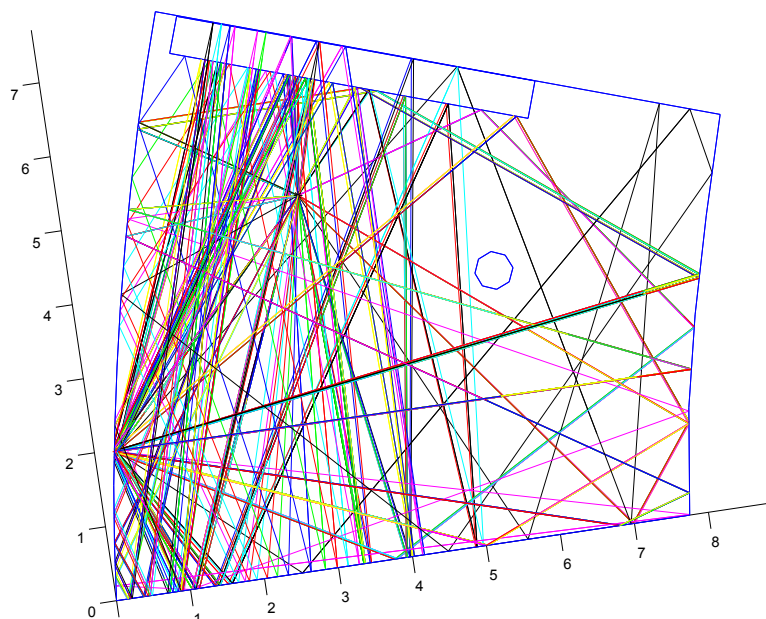


Figura 5.23: Simulación Ispace caso 3 con orden 5 de reflexión

totalmente circular en la realidad, generarán difracciones de borde. No obstante, esto es sencillo de solucionar ya que se prescindiría de una simulación en la que se tengan en cuenta difracciones en bordes. Asimismo, según se puede observar tanto en la Figura 5.11 como en la 5.20, entre otras, aunque exista un obstáculo pegado a una pared, el sistema verá un borde donde puede existir difracción y marcará rayos entre el obstáculo y la pared. No obstante, este aspecto no es perjudicial, sobretodo en la primera de las figuras ya que suele existir separación entre el obstáculo y la pared.

No obstante, cabe destacar que los resultados obtenidos con la herramienta de simulación permiten obtener trayectorias tanto en recintos paralelepípedos como en recintos con geometrías arbitrarias, siempre teniendo en cuenta las particularidades de la simulación de la difracción en bordes. Además, es posible seleccionar tanto la posición de la fuente como del receptor, así como el orden de reflexión como de difracción entre otros parámetros. Por tanto, la herramienta de simulación implementada cumple con todos los objetivos deseados en un principio.



## Capítulo 6

# Conclusiones y líneas futuras

### 6.1 Introducción

En este capítulo se especificarán aquellas conclusiones generales derivadas del desarrollo de este [TFM](#) y se propondrán una serie de líneas futuras de investigación derivadas de éste.

### 6.2 Conclusiones

La acústica supone un campo de estudio muy extenso, debido principalmente a la multitud de científicos y de estudios diferentes a lo largo de los siglos. Aun con todo esto, sigue siendo un área en auge por la introducción a éste de las nuevas tecnologías surgidas, como por ejemplo la aplicación de procesadores cada vez más potentes que permitan tiempos de respuesta muy bajos, llegando así a realizar simulaciones u otro tipo de estudios en tiempo real. Por tanto, la realización de un trabajo en este área supone la necesidad de conocer multitud de teorías ya desarrolladas. Como se ha explicado en este documento, a la hora de realizar un sistema de simulación acústica hay que tener en cuenta varios aspectos clave, como puede ser la elección del método o de la teoría de simulación a aplicar para el tipo de problema que se desee resolver. En este caso se deseaba conocer los trayectos seguidos por las ondas acústicas en un recinto, y la elección de la teoría ha sido aquella que permite modelar las ondas acústicas como rayos.

La existencia de numerosas herramientas de simulación acústica hacen disponer de una gran variedad de métodos de simulación ya implementados. La mayor parte de las herramientas más potentes se tratan de programas con licencias propietarias, de las cuales no se puede comprobar el código fuente y servir de ayuda en la generación de un sistema de simulación propio adaptado al problema concreto. Por su parte, cabe destacar que existe un gran número de programas con licencia libre generados por diversos investigadores en el campo de la acústica para comprobar las nuevas teorías desarrolladas. El principal inconveniente de éstos es que no tienen ninguna garantía de un funcionamiento correcto, aunque como se ha podido comprobar, existen varios con una buena calidad. En este caso, tras una laboriosa tarea de investigación de herramientas, ha existido la posibilidad de generar el nuevo sistema de simulación a partir de una herramienta ya desarrollada con código abierto, y que además incluye el cálculo de difracciones en bordes, siendo este un aspecto poco común en la mayoría de sistemas de simulación.

De este modo se ha implementado un sistema de simulación al que se puede acceder a través de una interfaz gráfica de usuario, así como desde una interfaz de programación de aplicaciones. La primera de ellas permite un manejo más sencillo para usuarios menos experimentados en el campo de la programación, así como la representación de los resultados de la simulación de forma gráfica en el recinto. La

interfaz de programación está destinada a facilitar la integración de este sistema de simulación con otros desarrollos. Uno de los principales escollos a superar, ha sido la posibilidad de introducir al recinto geometrías definidas en `.env/.srf`, realizando diversos desarrollos que permitieran la definición de superficies con el orden de vértices aceptado por la herramienta EDT. Otro de los objetivos principales es que el sistema implementado sea capaz de simular recintos no solo con geometría paralelepípeda como suele suceder en la mayor parte de herramientas disponibles, sino con cualquier geometría, aspecto conseguido en el sistema implementado. Es posible la introducción al sistema por parte del usuario del mismo, las posiciones de las fuentes y receptores, así como algunos otros parámetros de simulación, aportando así una flexibilidad añadida al mismo.

Los resultados obtenidos tras la ejecución de la herramienta de simulación son los esperados, constituyendo por tanto este sistema implementado una buena opción para el objetivo con el que fue creado. Hay que tener en cuenta a la hora de su utilización dos aspectos que pueden llevar a resultados no deseados, el primero de ellos es la definición correcta de los vértices de las superficies en geometrías no paralelepípedas, y el segundo la existencia de bordes que generan difracciones en lugares no deseados. En conclusión, se han conseguido y en algunos aspectos superado los objetivos previstos antes del desarrollo del sistema de simulación acústica.

### 6.3 Líneas futuras

Se proponen a continuación algunas líneas futuras de investigación para mejorar el sistema de simulación desarrollado en este TFM:

- Incluir en el sistema de simulación el cálculo de respuestas al impulso de difracción bajo un método diferente al implementado. También se propone incluir en el sistema de simulación el cálculo de las funciones de transferencia, obteniendo así una respuesta en el dominio frecuencial además de en el temporal.
- Realización de un algoritmo que permita la definición de cada superficie a través de sus vértices teniendo en cuenta el sentido de las normales correcto para recintos y obstáculos no paralelepípedos.
- La herramienta EDT empleada en el sistema de simulación propone una serie de mejoras que se pueden realizar, entre ellas se destaca el uso de un método geométrico diferente como puede ser trazado de rayos o trazado de haz, que aportan ventajas con respecto al método fuente imagen.
- Generar un sistema de simulación basado en tiempo real que contemple el movimiento de las posibles fuentes y receptores en el recinto.



## Capítulo 7

# Presupuesto

### 7.1 Costes de equipamiento

- Equipamiento hardware:

Concepto	Cantidad	Coste Unitario	Subtotal
PC Intel® Core™ 2 Duo CPU P7350	1	600 €	600 €
<b>Coste total HW</b>			<b>600 €</b>

Tabla 7.1: Coste del equipamiento hardware utilizado

- Recursos software:

Concepto	Cantidad	Coste Unitario	Subtotal
Windows 7 Ultimate	1	150 €	150 €
Matlab R2014a	1	8.000 €	8.000 €
TeXnicCenter	1	0	0 €
<b>Coste total SW</b>			<b>8.150 €</b>

Tabla 7.2: Coste del equipamiento software utilizado

### 7.2 Costes de mano de obra

Concepto	Cantidad	Coste Unitario	Subtotal
Estudio inicial	80 horas	30 €/hora	2400 €
Desarrollo SW	160 horas	70 €/hora	11200 €
Realización del documento	90 horas	25 €/hora	2250 €
<b>Coste total mano de obra</b>			<b>15.850 €</b>

Tabla 7.3: Coste de mano de obra

Concepto	Subtotal
Coste total HW	600 €
Coste total SW	8.150 €
Coste total mano de obra	15.850 €
<b>Coste total presupuesto</b>	<b>24.600 €</b>

Tabla 7.4: Coste total del presupuesto

### 7.3 Coste total del presupuesto

El importe total del presupuesto asciende a la cantidad de VEINTICUATRO MIL SEISCIENTOS €.

# Bibliografía

- [1] F. C. Tommasini, “Sistema de simulación acústica virtual en tiempo real,” Ph.D. dissertation, 2012.
- [2] C. H. Macías, “Estudio Acústico del Teatro Tomás y Valiente de Fuenlabrada,” Master’s thesis, Universidad Politécnica de Madrid, 2013.
- [3] Wikipedia, “Acústica— Wikipedia, La enciclopedia libre,” <https://es.wikipedia.org/wiki/Ac%C3%BAstica> [Último acceso 03/agosto/2016].
- [4] —, “Acústica arquitectónica— Wikipedia, La enciclopedia libre,” [https://es.wikipedia.org/wiki/Ac%C3%BAstica\\_arquitect%C3%B3nica](https://es.wikipedia.org/wiki/Ac%C3%BAstica_arquitect%C3%B3nica) [Último acceso 03/agosto/2016].
- [5] —, “Auralización— Wikipedia, La enciclopedia libre,” <https://es.wikipedia.org/wiki/Auralizaci%C3%B3n> [Último acceso 03/agosto/2016].
- [6] J. Velasco, D. Pizarro, and J. Macias-Guarasa, “Source localization with acoustic sensor arrays using generative model based fitting with sparse constraints,” *Sensors*, vol. 12, no. 10, pp. 13 781–13 812, 2012.
- [7] J. Velasco, C. J. Martín-Arguedas, J. Macias-Guarasa, D. Pizarro, and M. Mazo, “Proposal and validation of an analytical generative model of srp-phat power maps in reverberant scenarios,” *Signal Processing*, vol. 119, pp. 209–228, 2016.
- [8] J. Velasco, D. Pizarro, J. Macias-Guarasa, and A. Asaei, “Tdoa matrices: Algebraic properties and their application to robust denoising with missing data,” *IEEE Transactions on Signal Processing*, vol. 64, no. 20, pp. 5242–5254, Oct 2016.
- [9] R. Macho-Pedroso, F. Domingo-Perez, J. Velasco, C. Losada-Gutierrez, , and J. Macias-Guarasa, “Optimal microphone placement for indoor acoustic localization using evolutionary optimization,” in *Proceedings of the 2016 Indoor Positioning and Indoor Navigation Conference*, Alcalá de Henares, SPAIN, 2016, p. To appear.
- [10] H. Kuttruff, *Room acoustics*. London, New York: Spon Press, 2000. [Online]. Available: <http://opac.inria.fr/record=b1097189>
- [11] L. L. Beranek and D. T. Blackstock, “Acoustics, 1986 edition by Leo L. Beranek,” *The Journal of the Acoustical Society of America*, vol. 83, no. 3, pp. 1206–1207, 1988. [Online]. Available: <http://scitation.aip.org/content/asa/journal/jasa/83/3/10.1121/1.396027>
- [12] M. Taylor, A. Chandak, L. Antani, and D. Manocha, “Interactive geometric sound propagation and rendering.”
- [13] Wikipedia, “Reflexión(sonido)— Wikipedia, La enciclopedia libre,” [https://es.wikipedia.org/wiki/Reflexi%C3%B3n\\_\(sonido\)](https://es.wikipedia.org/wiki/Reflexi%C3%B3n_(sonido)) [Último acceso 03/agosto/2016].

- [14] —, “Difracción(sonido) — Wikipedia, La enciclopedia libre,” [https://es.wikipedia.org/wiki/Difracci%C3%B3n\\_\(sonido\)](https://es.wikipedia.org/wiki/Difracci%C3%B3n_(sonido)) [Último acceso 03/agosto/2016].
- [15] —, “Kirchhoffs diffraction formula— Wikipedia, La enciclopedia libre,” [https://en.wikipedia.org/wiki/Kirchhoff%27s\\_diffraction\\_formula](https://en.wikipedia.org/wiki/Kirchhoff%27s_diffraction_formula) [Último acceso 08/septiembre/2016].
- [16] R. R. Torres, U. Svensson, and M. Peter Kleiner, “Edge diffraction in room acoustics computations.”
- [17] T. Lokki, P. Svensson, and L. Savioja, “An efficient auralization of edge diffraction,” 2002.
- [18] U. P. S. Rendell R. Torres and M. Kleiner, “Computation of edge diffraction for more accurate room acoustics auralization,” *The Journal of the Acoustical Society of America*, vol. 109, no. 2, pp. 600–610, 2001. [Online]. Available: <http://scitation.aip.org/content/asa/journal/jasa/109/2/10.1121/1.1340647>
- [19] U. P. Svensson, R. I. Fred, and J. Vanderkooy, “An analytic secondary source model of edge diffraction impulse responses,” *The Journal of the Acoustical Society of America*, vol. 106, no. 5, pp. 2331–2344, 1999.
- [20] Wikipedia, “Refracción(sonido)— Wikipedia, La enciclopedia libre,” [https://es.wikipedia.org/wiki/Refracci%C3%B3n\\_\(sonido\)](https://es.wikipedia.org/wiki/Refracci%C3%B3n_(sonido)) [Último acceso 03/agosto/2016].
- [21] M. R. Schroeder, “Novel uses of digital computers in room acoustics,” *Acoustical Society of America Journal*, vol. 33, p. 1669, 1961.
- [22] A. Krokstad, S. Strom, and S. Sarsdal, “Calculating the acoustical room response by the use of a ray tracing technique,” *Journal of Sound and Vibration*, vol. 8, no. 1, pp. 118 – 125, 1968. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0022460X68901983>
- [23] L. Savioja and U. P. Svensson, “Overview of geometrical room acoustic modeling techniques,” *The Journal of the Acoustical Society of America*, vol. 138, no. 2, pp. 708–730, 2015.
- [24] R. H. Lyon, R. G. DeJong, and M. Heckl, “Theory and application of statistical energy analysis, second edition,” *The Journal of the Acoustical Society of America*, vol. 98, no. 6, pp. 3021–3021, 1995. [Online]. Available: <http://scitation.aip.org/content/asa/journal/jasa/98/6/10.1121/1.413875>
- [25] Wikipedia, “Statistical energy analysis — Wikipedia, La enciclopedia libre,” [https://en.wikipedia.org/wiki/Statistical\\_energy\\_analysis](https://en.wikipedia.org/wiki/Statistical_energy_analysis) [Último acceso 03/agosto/2016].
- [26] C. Lauterbach, A. Chandak, and D. Manocha, “Interactive sound rendering in complex and dynamic scenes using frustum tracing,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1672–1679, 2007.
- [27] J. B. Allen and D. A. Berkley, “Image method for efficiently simulating small room acoustics,” *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979. [Online]. Available: <http://scitation.aip.org/content/asa/journal/jasa/65/4/10.1121/1.382599>
- [28] J. Borish, “Extension of the image model to arbitrary polyhedra,” *J. Acoust. Soc. Am*, vol. 75, no. 6, pp. 1827–1836, 1984.
- [29] Wikipedia, “Beam tracing— Wikipedia, La enciclopedia libre,” [https://en.wikipedia.org/wiki/Beam\\_tracing](https://en.wikipedia.org/wiki/Beam_tracing) [Último acceso 03/agosto/2016].
- [30] S. Laine, S. Siltanen, T. Lokki, and L. Savioja, “Accelerated beam tracing algorithm,” *Appl. Acoust.*, vol. 70, no. 1, pp. 172–181, 2009.

- [31] Wikipedia, “Cone tracing— Wikipedia, La enciclopedia libre,” [https://en.wikipedia.org/wiki/Cone\\_tracing](https://en.wikipedia.org/wiki/Cone_tracing) [Último acceso 03/agosto/2016].
- [32] U. Stephenson, “An acoustic computer simulation technique for calculating parameters relevant to subjective acoustical impression in concert halls,” 1985.
- [33] —, “Comparison of the mirror image source method and the sound particle simulation method,” 1990.
- [34] Wikipedia, “Método de los elementos finitos— Wikipedia, La enciclopedia libre,” [https://es.wikipedia.org/wiki/M%C3%A9todo\\_de\\_los\\_elementos\\_finitos](https://es.wikipedia.org/wiki/M%C3%A9todo_de_los_elementos_finitos) [Último acceso 08/agosto/2016].
- [35] —, “Finite element method— Wikipedia, La enciclopedia libre,” [https://en.wikipedia.org/wiki/Finite\\_element\\_method](https://en.wikipedia.org/wiki/Finite_element_method) [Último acceso 08/agosto/2016].
- [36] —, “Boundary element method — Wikipedia, La enciclopedia libre,” [https://en.wikipedia.org/wiki/Boundary\\_element\\_method](https://en.wikipedia.org/wiki/Boundary_element_method) [Último acceso 08/agosto/2016].
- [37] R. S. Enrique Navarro Camba, Jaume Segura, “Aplicación del método de Diferencias Finitas en el Dominio del Tiempo (FDTD) al estudio de problemas acústicos bidimensionales,” universidad de Valencia. [Online]. Available: <http://www.sea-acustica.es/revista/VOLXXXIV34/03.pdf>
- [38] Wikipedia, “Finite-difference time-domain method — Wikipedia, La enciclopedia libre,” [https://en.wikipedia.org/wiki/Finite-difference\\_time-domain\\_method](https://en.wikipedia.org/wiki/Finite-difference_time-domain_method) [Último acceso 03/agosto/2016].
- [39] U. P. d. M. ETSIST, “Acústica arquitectónica,” Diapositivas.
- [40] “Página de herramienta ABEC3,” <http://www.randteam.de/ABEC3/> [Último acceso 13/junio/2016].
- [41] “Página de herramienta AIST Laboratory,” <http://www.aist.aaanet.ru/> [Último acceso 13/junio/2016].
- [42] “Página de herramienta Aurora y Ramsete,” <http://www.aurora-plugins.com/> [Último acceso 13/junio/2016].
- [43] “Página de herramienta Cara,” <https://www.cara.de/> [Último acceso 13/junio/2016].
- [44] “Página de herramienta CATT-Acoustics,” <http://www.catt.se/> [Último acceso 13/junio/2016].
- [45] “Página de herramienta EASE,” <http://ease.afmg.eu/> [Último acceso 13/junio/2016].
- [46] “Página de herramienta RAYNOISE,” [http://www.plm.automation.siemens.com/es\\_es/products/lms/virtual-lab/legacy-applications/raynoise.shtml](http://www.plm.automation.siemens.com/es_es/products/lms/virtual-lab/legacy-applications/raynoise.shtml) [Último acceso 13/junio/2016].
- [47] “Página de herramienta ODEON,” <http://www.odeon.dk/> [Último acceso 13/junio/2016].
- [48] “Página de herramienta OlivetreeLab,” <http://www.olivetreeLab.com/Room> [Último acceso 13/junio/2016].
- [49] “Página de herramienta RAMSETE,” <http://www.ramsete.com/> [Último acceso 13/junio/2016].
- [50] P. D. T. Cox., “Room optimizer: A computer program to optimize the placement of listener, loudspeakers, acoustic surface treatment and room dimensions in critical listening room.”
- [51] “Página de herramienta para descarga RoomOptimizer,” <http://room-optimizer.software.informer.com/> [Último acceso 13/junio/2016].

- [52] “Página de herramienta RPlusD(ETF),” <http://www.etfacoustic.com/index.html> [Último acceso 13/junio/2016].
- [53] “Página de herramienta WinMLS,” <http://www.winmls.com/> [Último acceso 13/junio/2016].
- [54] “Página de herramienta Abersim,” <https://www.ntnu.edu/isb/ultrasound/abersim> [Último acceso 18/julio/2016].
- [55] “Página de herramienta AC2D,” <http://individual.utoronto.ca/kzhu/program/ac2d/doc/html/index.html> [Último acceso 18/julio/2016].
- [56] M. Queiroz, F. Iazzetta, F. Kon, M. A. H. A. Gomes, F. A. L. Figueiredo, B. Masiero, L. K. Ueda, L. Dias, M. A. H. C. Torres, and L. F. Thomaz, “AcMus: an open, integrated platform for room acoustics research,” *Journal of the Brazilian Computer Society*, vol. 14, pp. 87 – 103, 09 2008. [Online]. Available: [http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0104-65002008000300007&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0104-65002008000300007&nrm=iso)
- [57] “Página de herramienta Acoustic Simulator,” <https://github.com/thecodeboss/AcousticSimulator> [Último acceso: 18/julio/2016].
- [58] “Página herramienta AcousSTO,” <http://acousto.sourceforge.net/index.php> [Último acceso:18/julio/2016].
- [59] “Página de herramienta BEM++,” <http://www.bempp.org/> [Último acceso:18/julio/2016].
- [60] “Página de herramienta BEM Acoustics,” <http://www.bemacoustics.com/> [Último acceso:18/julio/2016].
- [61] “Página de herramienta Blender,” <https://www.blender.org/> [Último acceso:18/julio/2016].
- [62] “Página de herramienta CABRILLO,” <http://noiselab.ucsd.edu/cabrillo/cabrillo.html> [Último acceso:18/julio/2016].
- [63] “Página de herramienta Clawpack,” <http://www.clawpack.org/> [Último acceso:18/julio/2016].
- [64] “Página de herramienta CREANIUS,” <http://www.creatis.insa-lyon.fr/site/fr/CREANUIS.html> [Último acceso:18/julio/2016].
- [65] “Página de herramienta CodeTYMPAN,” <http://www.code-tympa.org/> [Último acceso 25/julio/2016].
- [66] “Página de herramienta the Dream MATLAB Toolbox,” <http://www.signal.uu.se/Toolbox/dream/> [Último acceso 24/julio/2016].
- [67] “Página de herramienta Edge Diffraction Toolbox,” <http://www.iet.ntnu.no/~svensson/software/> [Último acceso:18/julio/2016].
- [68] “Página de herramienta Elmer,” <http://www.elmerfem.org/> [Última acceso: 25/julio/2016].
- [69] “Página de herramienta EVERTims,” <http://evertims-manual.readthedocs.io/en/latest/> [Última acceso: 18/julio/2016].
- [70] “Página de herramienta FDAC3DMOD,” [https://sourceforge.net/projects/fdac3dmod/?source=typ\\_redi](https://sourceforge.net/projects/fdac3dmod/?source=typ_redi) [Último acceso:18/julio/2016].

- [71] “Página de herramienta Image-Sources,” <https://github.com/bmershon/image-sources> [Último acceso:18/julio/2016].
- [72] “Página de herramienta i-Simpa,” <http://i-simpa.ifttar.fr/> [Último acceso:18/julio/2016].
- [73] “Página de herramienta k-Wave,” <http://www.k-wave.org/index.php> [Último acceso 24/julio/2016].
- [74] “Página de herramienta KZK Bergen Code,” ([uib.no/people/nmajb/](http://uib.no/people/nmajb/)) [Último acceso 24/julio/2016].
- [75] “Página de herramienta KZK Texas Code,” [people.bu.edu/robinc/kzk/](http://people.bu.edu/robinc/kzk/) [Último acceso 24/julio/2016].
- [76] “Página de herramienta MIMPy,” <https://github.com/asgeran/MIMPy> [Último acceso 24/julio/2016].
- [77] “Página de herramienta NiHu,” <http://last.hit.bme.hu/nihu/> [Último acceso 25/julio/2016].
- [78] “Página de herramienta Open Acoustics,” <http://www.openacoustics.org/> [Último acceso 24/julio/2016].
- [79] “Página de herramienta OpenPSTD,” <http://www.openpstd.org/> [Último acceso 24/julio/2016].
- [80] “Página de herramienta Pachyderm Acoustical Simulation,” <http://www.perspectivesketch.com/pachyderm/> [Último acceso 24/julio/2016].
- [81] “Página de herramienta Pachyderm Acoustical Simulation,” <https://github.com/PachydermAcoustic> [Último acceso 24/julio/2016].
- [82] “Página de herramienta Parallel FDTD,” <https://github.com/juuli/ParallelFDTD> [Último acceso 24/julio/2016].
- [83] “Página de herramienta Pogo,” <http://www.pogo-fea.com/> [Último acceso 24/julio/2016].
- [84] “Página de herramienta RoomAcoustics,” <https://github.com/MuffinUprising/RoomAcoustics> [Último acceso 24/julio/2016].
- [85] “Página de herramienta Room Acoustic Simulation with CUDA,” [https://github.com/ws2367/Room\\_Acoustic\\_Simulation\\_with\\_CUDA](https://github.com/ws2367/Room_Acoustic_Simulation_with_CUDA) [Último acceso 24/julio/2016].
- [86] D. R. Campbell, K. J. Palomaki, and G. Brown, “A MATLAB simulation of shoebox room acoustics for use in research and teaching.” *Computing and Information Systems Journal*, ISSN 1352-9404, vol. 9, no. 3, 2005.
- [87] “Página de herramienta Simsonic,” <http://www.simsonic.fr/> [Último acceso 24/julio/2016].
- [88] “Página de herramienta Slab3d,” <http://slab3d.sourceforge.net/> [Último acceso 24/julio/2016].
- [89] “Página de herramienta Ultrasim,” <http://www.mn.uio.no/ifi/english/research/groups/dsb/resources/software/ultrasim/index.html> [Último acceso 24/julio/2016].
- [90] “Página de herramienta Ultrasim (Mathworks),” <https://www.mathworks.com/matlabcentral/fileexchange/1335-ultrasim> [Último acceso 24/julio/2016].
- [91] J. Sheaffer and Fazenda, “Wavecloud: an open source room acoustics simulator using the finite difference time domain method,” *Forum Acusticum*, 2014.

- [92] “Página personal Peter Svensson,” <http://www.iet.ntnu.no/~svensson/index.html> [Último acceso 7/septiembre/2016].
- [93] “Página de herramientas organizativas,” <http://www.freeacoustics.com/> [Último acceso 13/junio/2016].
- [94] “Página de entorno de desarrollo de GUI de Matlab,” <http://es.mathworks.com/discovery/matlab-gui.html> [Último acceso 12/septiembre/2016].
- [95] “Mathworks, intersección recta y plano,” <https://es.mathworks.com/matlabcentral/fileexchange/24484-geom3d/content/geom3d/geom3d/intersectLinePlane.m> [Último acceso 13/septiembre/2016].
- [96] Wikipedia, “Software propietario— wikipedia, la enciclopedia libre,” [https://es.wikipedia.org/wiki/Software\\_propietario](https://es.wikipedia.org/wiki/Software_propietario) [Último acceso 22/agosto/2016].
- [97] “Artículo: "tipos de licencia de software",” <http://www.informatica-hoy.com.ar/software-libre-gnu/Tipos-de-licencia-de-Software.php> [Último acceso:20/agosto/2016].
- [98] Wikipedia, “Software libre— wikipedia, la enciclopedia libre,” [https://es.wikipedia.org/wiki/Software\\_libre](https://es.wikipedia.org/wiki/Software_libre) [Último acceso 22/agosto/2016].
- [99] “Artículo: "las 5 licencias de software libre más importantes que todo desarrollador debe conocer",” <https://bbvaopen4u.com/es/actualidad/las-5-licencias-de-software-libre-mas-importantes-que-todo-desarrollador-debe-conocer> [Último acceso:20/agosto/2016].



# Apéndice A

## Licencias de software

Una licencia de software es una autorización formal con carácter contractual que el autor de un programa otorga a un interesado para ejercer actos de explotación legales. Es decir, éstas no son más que un contrato entre el autor del software y el usuario que utiliza el mismo, estableciendo una serie de términos y cláusulas que el usuario debe cumplir para utilizarlo.

Durante la década de los 60 el software no se consideraba un producto, sino un añadido que los vendedores de computadores de la época aportaban a los clientes para que pudieran usarlos, por tanto, los autores de los programas los compartían libremente. En la década de los 70, la informática no había sufrido todavía su gran auge y existía principalmente en ámbitos universitarios y empresariales, donde se creaba y compartía el software sin ningún tipo de restricción. No fue hasta comienzos de los 80 cuando la situación comenzó a cambiar coincidiendo con la introducción de la informática a nivel de usuario pues las compañías tecnológicas comenzaron a restringir el uso de su software imponiendo restricciones con el uso de acuerdos de licencia. Es en este momento cuando se introducen sistemas operativos privativos, forzando a los usuarios a aceptar condiciones restrictivas que impedían realizar modificaciones al software.

En la actualidad, se distinguen dos grandes tipos de licencias que se detallarán a continuación: las licencias de software propietario y las de software libre.

### A.1 Software propietario

El software propietario o privativo [96] implica que el uso, redistribución o modificación no está permitido o requiere el permiso del titular del software. En este tipo de software no se tiene libertad de acceso al código fuente.

Existen multitud de licencias de software propietario ya que cada desarrollador o autor, generará normalmente la suya propia. No obstante, se destacan una serie de términos genéricos que a efectos de la selección realizada en este trabajo han sido considerados como software propietario [97]:

- **Freeware:** del inglés *free software*, es un tipo de software que permite la redistribución pero no la modificación, y que en ocasiones incluye el código fuente aunque no es usual este aspecto. Se distribuye sin coste, disponible para su uso y por tiempo ilimitado. Un programa freeware es un programa gratuito pero no es libre, es una variante gratuita del software de tipo shareware. Suele incluir una licencia de uso, que permite la redistribución con restricciones como no modificar la aplicación ni venderla, y dar cuenta al autor. El autor puede desautorizar el uso de su software a una compañía con fines comerciales o a una entidad gubernamental, o puede requerir pagos si el

uso va a ser comercial. Un software freeware es frecuente que permita sólo el uso no comercial, o uso académico, o una combinación entre varios permisos.

- **Shareware:** licencia software en la que el usuario puede evaluar de forma gratuita el producto pero con limitaciones en el tiempo o forma de uso, o con restricciones en las capacidades finales. Puede ser redistribuido pero su utilización implica el pago, de esta forma se puede obtener el registro del programa o la versión integral con todos los recursos. Normalmente el código fuente no se encuentra disponible y por tanto no se pueden realizar modificaciones. Abarca las licencias adware, trial y demo que se detallan a continuación:
  - **Adware:** subprograma que descarga publicidad sobre otro programa principal. Ocurre cuando un programa tiene versiones comerciales o más avanzadas que necesitan ser compradas para poder ser utilizadas. Pagando por la versión comercial los anuncios desaparecen.
  - **Trial:** versión de programa pago, distribuido gratuitamente con todos los recursos activos pero por un tiempo determinado. Es como una licencia freeware, que después de un tiempo deja de funcionar. Para continuar utilizándolo se debe comprar la clave de registro.
  - **Demo:** versión de demostración, liberada por el autor, que no contiene todas las funciones del programa original. Es distribuida gratuitamente, y no tiene plazo de validez, pero también tiene pocos recursos y funciones.

## A.2 Software libre

El software libre [98], *free software* en inglés, implica que el usuario que lo utilice puede copiarlo, estudiarlo, modificarlo, utilizarlo libremente para cualquier fin y redistribuirlo con o sin cambios o mejoras. La definición de este software está asociada al nacimiento del movimiento de software libre, encabezado por Richard Stallman y la consecuente fundación aparecida en 1985 *Free Software Foundation*.

Según la definición establecida por Richard Stallman, el software es libre si garantiza estas cuatro libertades (ver esquema A.1):

- Libertad 0: USAR el programa con cualquier propósito.
- Libertad 1: ESTUDIAR como funciona y tener la opción de modificarlo adaptándolo a las propias necesidades.
- Libertad 2: DISTRIBUIR copias del programa para poder ayudar a otros usuarios.
- Libertad 3: MEJORAR el programa y hacer públicas las mejoras de modo que toda la comunidad se beneficie.

Las libertades 1 y 3 requieren de acceso al código fuente.

Las dos motivaciones del software libre pueden definirse como:

1. Motivación ética: abanderada por la *Free Software Foundation* (partidaria de la denominación como fuente libre), que argumenta que el software es conocimiento, debe poderse difundir sin trabas, su ocultación es una actitud antisocial y que la posibilidad de modificar programas es una forma de libertad de expresión.
2. Motivación pragmática: abanderada por la *Open Source Initiative* (partidaria de la denominación como fuente abierta), que argumenta ventajas técnicas y económicas, apartando el término free para poder evitar así la posible confusión entre libre y gratis.

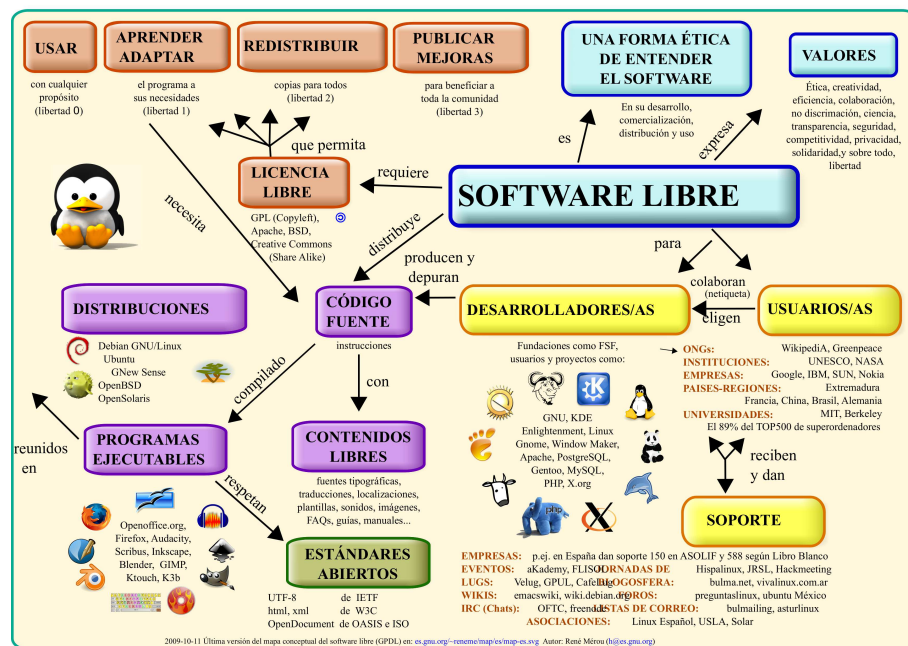


Figura A.1: Mapa conceptual del software libre (Fuente: es.gnu.org)

El término *free* en lengua inglesa presenta una ambigüedad entre el significado de libre o gratis. Por tanto, se diferencia entre software libre y gratuito, el primero puede ser gratuito o de pago, pero el usuario siempre será libre de usarlo como desee, mientras que el software gratuito (freeware) a pesar de ser gratuito puede estar restringido por algún tipo de licencia con la que el usuario no es libre de actuar. La libertad tiene que ver con el uso y no con la gratuidad.

La principal ventaja de los programas bajo licencia libre es que permite a millones de programadores aplicar su conocimiento sobre el software, contribuyendo de este modo a mejorarlo. A la vez que se aplica el conocimiento se amplía el mismo, por tanto, la disponibilidad de este tipo de software permite la expansión del conocimiento. Una de sus principales ventajas es que el hecho de acceder al código fuente proporciona una cierta independencia tecnológica, pudiendo desarrollar tu propia versión.

A continuación se destacan algunas de las principales licencias de software libre [99]:

- **GPL:** denominada normalmente como GNU GPL, el desarrollador conserva los derechos de autor, pero permite su libre distribución, modificación y uso. Es necesario que si existe una modificación, el nuevo software quede obligatoriamente con la misma licencia, es decir, estás obligado a facilitar el código fuente del software modificado. El código fuente debe estar disponible y accesible, para copias ilimitadas y para cualquier persona que lo solicite. Para el usuario final, el software con esta licencia es totalmente gratuito, pudiendo pagar únicamente por gastos de copiado y distribución.
- **Lesser General Public License (LGPL):** Licencia Pública General Reducida de GNU, más conocida en inglés como GNU Lesser General Public License, es una licencia que garantiza la libertad de compartir y modificar el software, asegurando que el software es libre para todos los usuarios. Únicamente permite la copia, distribución y modificación, con la principal diferencia con la licencia **GPL** en que esta puede enlazarse a un programa sin licencia **GPL** (libre o privativo).
- **Affero General Public License (AGPL):** la licencia Affero GPL está destinada a modificar el derecho de autor de las licencias bajo GNU. Aparte de las cláusulas propias de la licencia GPL, obliga a que

se distribuya el software que se destine a dar servicios a través de una red de ordenadores, es decir, si se quiere usar como parte del desarrollo de un nuevo software, este quedaría obligado a su libre distribución. Existe la obligación de distribuir el software si este se ejecuta para ofrecer servicios a través de una red de ordenadores.

- *Berkeley Software Distribution* (BSD): no impone casi condiciones sobre lo que un usuario puede hacer con el software. El software bajo esta licencia es la menos restrictiva para los desarrolladores ya que el software puede ser vendido y no hay obligación de incluir el código fuente. Una aplicación con esta licencia permite que otras versiones puedan tener otros tipos de licencias, tanto libres como propietarias. El autor mantiene la protección de copyright únicamente para la renuncia de garantía y para requerir la adecuada atribución de la autoría en trabajos derivados, pero permite la libre redistribución y modificación, incluso si los trabajos tienen propietario. Se considera como la licencia verdadera de software libre porque el usuario tiene libertad ilimitada con respecto al software, pudiendo decidir incluso redistribuirlo como no libre.
- *Massachusetts Institute of Technology* (MIT): licencia originada en el Instituto Tecnológico de Massachusetts (MIT) y que más correctamente debería ser denominada licencia X11. Permite reutilizar software propietario y es compatible con la licencia GNU GPL, siendo posible integrar software con licencia MIT en software con licencia GNU GPL, pero no en caso contrario. También permite la modificación del código fuente y es bastante similar a la licencia [BSD](#).
- *Mozilla Public License* (MPL): la licencia pública de Mozilla está desarrollada y mantenida por la Fundación Mozilla. Se puede considerar una mezcla entre la licencia [BSD](#) modificada y la licencia [GPL](#), buscando equilibrar los intereses de los desarrolladores de código abierto y propietario. Permite que el código fuente pueda mezclarse con otros archivos bajo licencia incluso privativa. Aplica licencias dobles al código fuente y a los ejecutables, obligando a devolver al autor los códigos fuente modificados, aunque permite licenciar los binarios como propietarios.
- *Eclipse Public License* (EPL): conocida como Licencia Pública Eclipse, es utilizada por la Fundación Eclipse para licenciar su software. Diseñada para ser una licencia software con miras a fines empresariales, disponiendo de condiciones más débiles que las licencias GNU GPL. Permite usar, modificar, copiar y distribuir el software y las versiones modificadas, aunque en ocasiones se obliga a liberar los cambios realizados. No es compatible con la licencia GPL, por tanto, no está permitido crear software licenciado con ambas licencias.
- Apache: licencia creada por la fundación Apache Software (ASF), requiere la conservación del aviso de derechos de autor y el descargo de responsabilidad, pero no es copyleft, ya que no requiere la redistribución del código fuente cuando se distribuyen versiones modificadas. Se permite usar, modificar y distribuir las versiones modificadas del software. La licencia *Educational Community License* (ECL) se engloba dentro de este tipo de licencia.
- *CEA CNRS INRIA Logiciel Libre* (CeCILL): es una licencia francesa de software libre adaptada tanto a las leyes francesas como a los distintos tratados internacionales, además, es compatible con la licencia GNU GPL. Destaca dentro de esta licencia la conocida como CeCILL-B que es compatible con las licencias de tipo BSD (X11, MIT), disponiendo así de un requisito de atribución que normalmente no se permite en GPL.

# Apéndice B

## Manual de usuario

### B.1 Introducción

En este [TFM](#) se ha realizado una aplicación a través del entorno de desarrollo [GUI](#) de Matlab donde es posible introducir una serie de parámetros de la simulación por parte del usuario de la misma, así como observar los resultados representados gráficamente según las distintas opciones disponibles. Además, se ha realizado una interfaz de programación de aplicaciones ([API](#)) compuesta por dos funciones y que devuelve al usuario los resultados de la simulación en forma de matrices y estructuras de Matlab.

Por tanto, en este apéndice se explicará la forma de utilización de ambas aplicaciones.

### B.2 Manual de la interfaz gráfica

La aplicación generada con el entorno de desarrollo [GUI](#) de Matlab consta de tres interfaces; la primera de ellas dedicada a la portada de bienvenida a la aplicación, la segunda dedicada a la introducción de parámetros de simulación por parte del usuario, y la tercera dedicada a la visualización de los resultados. A continuación se describen los pasos necesarios para el correcto funcionamiento de la aplicación, desgranando el significado de las distintas opciones.

#### B.2.1 Interfaz de bienvenida

Ejecutando en Matlab el archivo de nombre `simulador1` aparecerá una pantalla como la que se muestra en la Figura [B.1](#). Para acceder a la siguiente interfaz tan sólo es necesario presionar con el ratón el botón inferior 'Entrar'.

#### B.2.2 Interfaz de introducción de parámetros

En la Figura [B.2](#) se muestra la pantalla que se observa tras pulsar el botón de la interfaz anterior. Cabe destacar que también es posible ejecutar desde Matlab directamente esta interfaz, tan sólo iniciando la simulación desde el archivo de nombre `simulador2`. En la presente interfaz es posible introducir información y seleccionar algunos parámetros con los que se quiere realizar la simulación.

En los siguientes apartados se especifica el significado de cada parámetro, así como el orden en el que se recomienda introducir las distintas opciones de esta interfaz.



Figura B.1: Pantalla de bienvenida al interfaz gráfico

Archivos	Opciones de simulación
<p>Su recinto se encuentra definido en la siguiente ubicación:</p> <p>Seleccione el archivo .ENV o .CAD <input type="button" value="Examinar"/></p> <p>Los archivos generados en la simulación comenzarán por la palabra:</p> <p><input type="text" value="20160916_090151"/></p> <p>Los archivos generados en la simulación se guardarán en:</p> <p>Seleccione el directorio donde se guardarán los archivos <input type="button" value="Examinar"/></p>	<p><b>Modelo</b></p> <p><input checked="" type="radio"/> Cerrado <input type="radio"/> Abierto</p> <p><b>Modelo</b></p> <p><input checked="" type="radio"/> Interior <input type="radio"/> Exterior</p> <p><input type="button" value="Ayuda"/></p> <p>Cantidad de texto a mostrar: <input type="text" value="4"/></p> <p><b>Condición de finalización:</b></p> <p>Orden máximo de reflexión especular: <input type="text" value="2"/></p> <p>Orden máximo de difracción en bordes: <input type="text" value="1"/></p> <p><b>Respuesta al impulso</b></p> <p><input type="checkbox"/> Calcular la respuesta al impulso</p> <p>Rstart [m]: <input type="text" value="1.3"/></p> <p>Frecuencia de muestreo: <input type="text" value="96000"/></p> <p>Velocidad del sonido [m/s]: <input type="text" value="344"/></p> <p>Densidad del aire [kg/m3]: <input type="text" value="1.21"/></p>
<p><b>Fuentes y receptores</b></p> <p>Introduzca el número de fuentes sonoras existentes: <input type="text" value="1"/></p> <p>Introduzca las coordenadas de la ubicación de las fuentes [metros]:</p> <p><input type="text"/> Formato: X1 Y1 Z1;X2 Y2 Z2;...</p> <p>Introduzca el número de receptores existentes: <input type="text" value="1"/></p> <p>Introduzca las coordenadas de la ubicación de los receptores [metros]:</p> <p><input type="text"/> Formato: X1 Y1 Z1;X2 Y2 Z2;...</p>	<p><input type="button" value="Visualizar"/> <input type="button" value="Simular"/></p>

Figura B.2: Pantalla de introducción de parámetros del interfaz gráfico

### B.2.2.1 Archivos

En este bloque que se observa en la Figura B.3 es posible rellenar tres campos de datos, a saber:

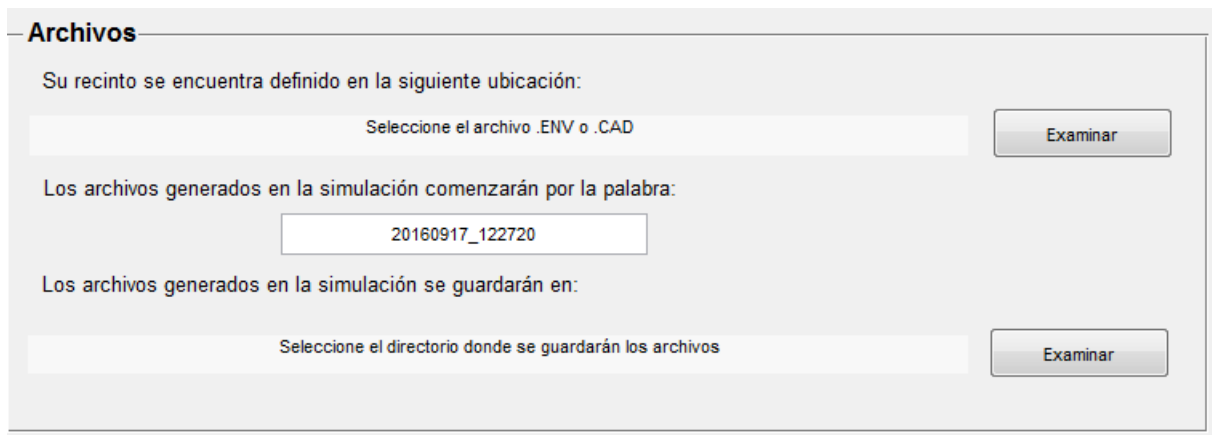


Figura B.3: Ampliación de la interfaz de introducción de parámetros relativos a archivos.

1. Pulsando el botón 'Examinar' superior aparecerá una ventana de archivos donde se podrá seleccionar el archivo con extensión `.env` o `.cad` que contiene la información relativa al recinto y los obstáculos donde se quiere realizar la simulación.
2. En el segundo campo se inserta la palabra con la que se desea comiencen los archivos generados en la simulación. Por defecto aparecerá la palabra formada por la fecha y la hora del momento en el que se ha ejecutado la aplicación con el formato `YYYYMMDD_HHMMSS`.
3. Pulsando el botón interior 'Examinar' aparecerá una ventana donde se puede seleccionar el directorio donde se desea que los archivos que genere la simulación sean almacenados.

Tras la introducción de estos tres parámetros es posible la visualización del recinto seleccionado pulsando el botón 'Visualizar' situado en el margen inferior derecho.

### B.2.2.2 Fuentes y receptores

En el bloque 'Fuentes y receptores' que se puede observar en la Figura B.4 se debe introducir la información solicitada siguiendo este orden:

1. En primer lugar se escribirá el número de fuentes que se desean introducir para realizar la simulación. Cada una de éstas será tratada individualmente, generando así unos resultados para cada una de ellas.
2. En el segundo campo se debe introducir la posición o posiciones de las fuentes que se han especificado en el paso anterior. Es necesario introducir las coordenadas en el formato especificado, de lo contrario no se leerán de forma correcta. Se debe indicar en primer lugar la coordenada  $x$  separada por un espacio de la  $y$ , y esta a su vez separada de la  $z$  por otro espacio. Si se desean añadir más fuentes se separarán entre ellas mediante un punto y coma.
3. En el tercer campo se introduce el número de receptores con los que se quiere realizar la simulación. Cada uno, al igual que las fuentes, será tratado individualmente, obteniendo así resultados para cada par fuente-receptor.

- En el cuarto campo se introducen las coordenadas de la posición o posiciones del número de receptores especificado en el paso anterior, siguiendo el mismo formato explicado en el punto 2.

**Fuentes y receptores**

Introduzca el número de fuentes sonoras existentes:

Introduzca las coordenadas de la ubicación de las fuentes [metros]:  
  
 Formato: X1 Y1 Z1;X2 Y2 Z2;...

Introduzca el número de receptores existentes:

Introduzca las coordenadas de la ubicación de los receptores [metros]:  
  
 Formato: X1 Y1 Z1;X2 Y2 Z2;...

Figura B.4: Ampliación de la interfaz de introducción de parámetros relativos a fuentes y receptores.

Si una vez introducida la información de este bloque se presiona el botón 'Visualizar' se puede observar el recinto y la posición de las fuentes y receptores especificados en una nueva ventana. Las fuentes aparecerán con un círculo en color rojo y los receptores con un círculo en color verde.

### B.2.2.3 Opciones de simulación

En el bloque 'Opciones de simulación' que se observa en la Figura B.5 se seleccionan aquellas opciones relativas a la simulación. A continuación se detalla cada una de las mismas:

- Modelo: en el caso de esta aplicación destinada a la simulación de la propagación del sonido en recintos cerrados se debe seleccionar un modelo cerrado e interior, no obstante, el sistema puede extenderse a otro tipo de ámbitos donde se desee seleccionar otro tipo de modelo. Hay que tener en cuenta que un modelo definido como interior siempre debe ser cerrado, en caso contrario se emitirá un mensaje de error emplazando a seleccionar esta opción.
- Condición de finalización: la simulación emplea el método geométrico de fuente imagen, por tanto es necesario incluir una condición de fin para que termine la misma. En este sistema se ha escogido una condición de fin especificada por los órdenes de reflexión y/o difracción que se deseen calcular. El orden de difracción especificado no puede superar al de reflexión, además, si el orden de reflexión es mayor o igual que 3 se debe seleccionar un orden de difracción 0.
- Respuesta al impulso: en esta sección se incluye en primer lugar un *checkbox* en el que si se encuentra activado se calculará en la simulación la respuesta al impulso. Los siguientes cuatro parámetros de este bloque únicamente deben ser modificados si se desea calcular la respuesta al impulso. Cada uno de estos tiene el siguiente significado:
  - Rstart: esta es una variable usada en la simulación cuyo fin es reducir el número de ceros al comienzo de la respuesta al impulso que aparecen en gran número si la distancia entre fuente y receptor es amplia y la frecuencia de muestreo elevada. En la ecuación B.1 se muestra la condición que debe cumplir en la simulación esta variable, siendo  $R$  la posición del receptor,  $S$  la posición de la fuente y  $norm$  la función de Matlab que calcula la norma. Por su parte



$CAIR$  representa la velocidad del sonido y  $FSAMP$  la frecuencia de muestreo, ambos valores se pueden seleccionar en los dos campos siguientes.

$$\frac{norm(R - S) - Rstart}{CAIR \cdot FSAMP} + 1 > 1 \quad (B.1)$$

- Frecuencia de muestreo: en este campo se indica la frecuencia de muestreo a la que se quiere calcular la respuesta al impulso, siendo introducido este valor en  $Hz$ .
  - Velocidad del sonido: en este campo se introduce la velocidad de propagación del sonido en el aire, siendo seleccionado por defecto la velocidad de este en condiciones normales.
  - Densidad del aire: en este último campo se introduce la densidad del aire del recinto que se esté simulando. El valor por defecto es el valor de la densidad en condiciones estándar.
4. En la parte superior derecha existe un desplegable donde se puede seleccionar la cantidad de texto a mostrar. Esta opción tiene que ver con el texto que se imprime en la ventana de comandos de Matlab y que se guardará en un archivo de texto accesible desde la tercera interfaz. A mayor valor, mayor cantidad de texto será imprimido. Se recomienda el valor por defecto para comprobar los distintos pasos de la simulación.

**Opciones de simulación**

**Modelo**  
☒ Cerrado  
☐ Abierto

**Modelo**  
☒ Interior  
☐ Exterior

**Ayuda**

Cantidad de texto a mostrar: 4

**Condición de finalización:**

Orden máximo de reflexión especular: 2

Orden máximo de difracción en bordes: 1

**Respuesta al impulso**

☐ Calcular la respuesta al impulso

Rstart [m]: 1.3

Frecuencia de muestreo: 96000

Velocidad del sonido [m/s]: 344

Densidad del aire [kg/m3]: 1.21

Figura B.5: Ampliación de la interfaz de introducción de parámetros relativos a las opciones de simulación.

En caso de existir alguna duda en la introducción de estos parámetros o en el significado de los mismos, existe un botón en amarillo de ayuda que contiene un resumen de las principales características de cada opción.

### B.2.3 Interfaz de visualización de resultados

Una vez haya concluido la simulación se abrirá una nueva interfaz como la que se observa en la Figura B.6 en la que se representarán las trayectorias obtenidas en la simulación para el recinto, obstáculos y fuentes y receptores definidos. Esta interfaz se divide en dos bloques principales, el bloque izquierdo donde se representará en 3D los resultados, y a la derecha las distintas opciones a seleccionar y que se detallan a continuación:

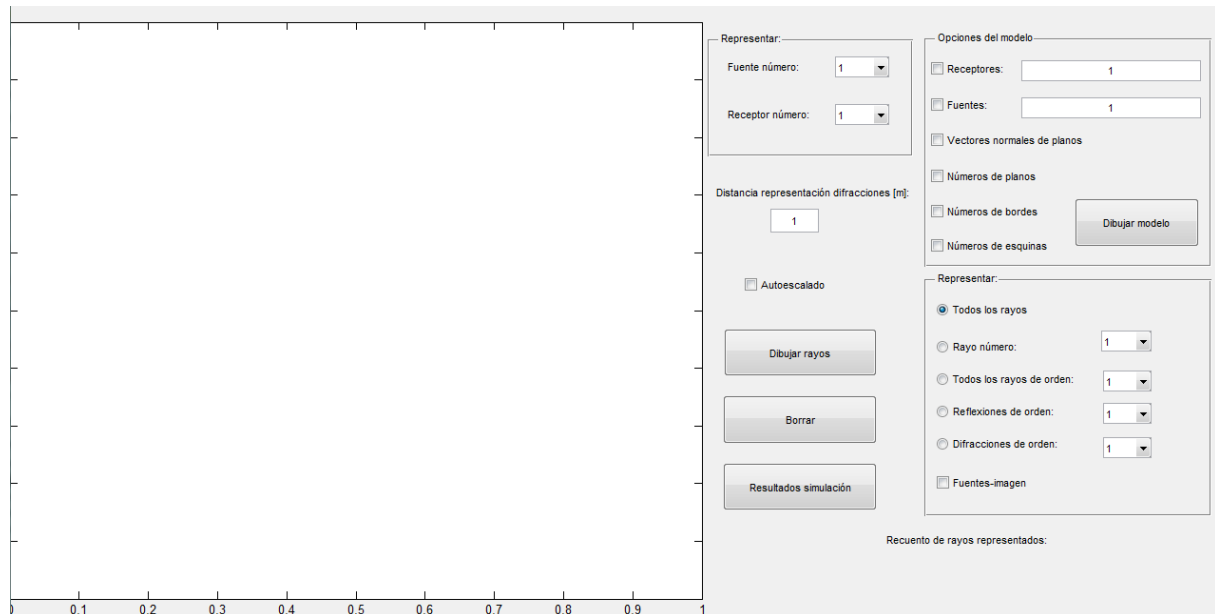


Figura B.6: Pantalla de visualización de la interfaz gráfica

1. En el bloque de la Figura B.7 se debe seleccionar el número de fuente y receptor a representar en el caso que se haya introducido más de uno en la interfaz anterior.

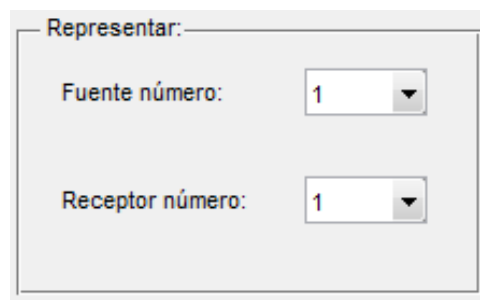


Figura B.7: Ampliación de la interfaz de visualización de resultados

2. El bloque de la Figura B.8 existe una serie de opciones que se pueden seleccionar al representar el recinto, a destacar:
  - Receptores y fuentes: en el caso de desear representar estas opciones, además de seleccionar el *checkbox* es necesario introducir en el recuadro el número de fuentes y receptores a representar en el caso de que exista más de uno. Esto es, si se han introducido dos receptores y deseamos representar ambos se escribirían los números 1 y 2 separados por un espacio. Si por el contrario, únicamente se desea representar el segundo receptor se escribiría un 2 en dicha casilla.

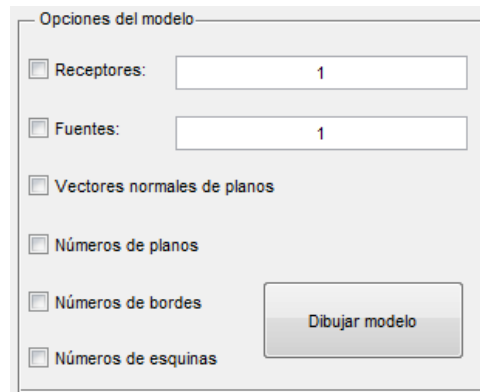


Figura B.8: Ampliación de la interfaz de visualización de resultados

- Vectores normales de planos: marcando esta opción se representarán unas líneas perpendiculares a cada superficie marcando el sentido en el que se han definido.
- Números de planos: marcando esta casilla, se representará el número de plano de cada superficie que forma la geometría.
- Números de bordes: en este caso se representará el número de borde que ha sido detectado tras la simulación en el punto medio de cada uno de ellos.
- Números de esquinas: marcando esta última opción se incluirá el número de cada uno de los vértices del recinto y obstáculos.

Una vez seleccionadas las opciones deseadas, se debe pulsar el botón 'Dibujar modelo' para que en el bloque izquierdo se represente tanto la geometría del recinto con sus obstáculos como las opciones seleccionadas. En el caso de que no se seleccione ninguna opción se dibujará únicamente el modelo.

3. En el bloque mostrado en la Figura B.9 es posible seleccionar una de las 5 opciones y pulsando el botón 'Dibujar rayos' se representará dicha opción en el bloque izquierdo. Acompañando a cada una de estas opciones se puede seleccionar o no la opción de representar las fuentes imagen que aparecerán con el símbolo de una estrella en color rojo. Hay que destacar que para visualizar tanto las opciones del bloque superior como las de este bloque es necesario pulsar el botón 'Dibujar modelo' y a continuación el botón 'Dibujar rayos'. A continuación se detallan distintas opciones de este bloque:

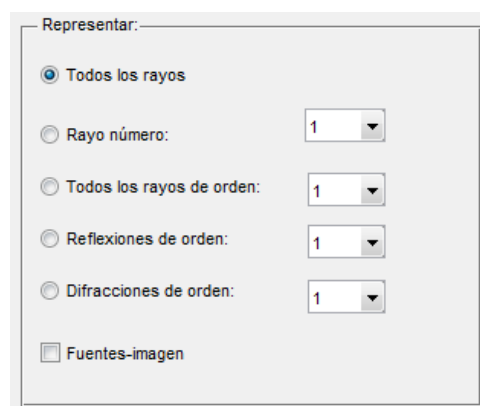


Figura B.9: Ampliación de la interfaz de visualización de resultados

- Todos los rayos: si se selecciona esta opción todos los rayos obtenidos en la simulación serán representados.

- Rayo número: Si se selecciona esta opción, se debe seleccionar en el *popupmenu* el número del rayo a representar. Para seguir representando rayos se debe seleccionar otro número de rayo y pulsar el botón 'Dibujar rayos'. Esto ocurrirá hasta que se pulse el botón 'Borrar' y se eliminen todas las opciones que se habían representado.
  - Todos los rayos de orden: representa tanto las reflexiones como las difracciones del orden seleccionado. Cabe destacar que en orden 1 se contabilizan los rayos con un orden de difracción y otro orden de reflexión, aunque el rayo en total tenga dos ordenes.
  - Reflexiones de orden: en este caso se representan únicamente las reflexiones especulares sin ninguna componente de difracción y del orden señalado en el *popupmenu* de su derecha.
  - Difracciones de orden: en este caso se contabiliza como orden 1 aquellos rayos con un orden de reflexión y otro de difracción. En orden 2 encontramos los rayos con dos difracciones.
4. En el bloque representado en la Figura B.10 existe la posibilidad de seleccionar la distancia existente entre puntos de un borde con el que se quieren representar rayos difractados, introduciendo la medida en metros. En el *checkbox* inferior es posible modificar el autoescalado de la representación en 3D.

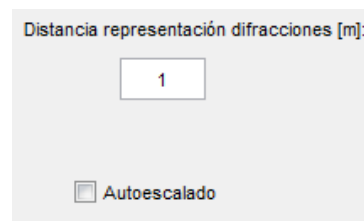


Figura B.10: Ampliación de la interfaz de visualización de resultados

5. En la parte inferior de la interfaz se mostrará un mensaje con información del número de trayectos diferentes representados según la opción elegida.
6. Por último cabe destacar el botón 'Resultados de simulación' que abre un bloc de notas en el que se muestran los resultados de la ejecución de la simulación.

## B.3 Manual de la interfaz de programación

Para utilizar este sistema de simulación acústica mediante la llamada a una función de Matlab, ha de realizarse de la siguiente forma:

1. Llamada a la función `InitSimulation()` con los valores deseados para cada uno de los parámetros que se especifican a continuación:
  - (a) `ENVfile`: hay que incluir en una cadena de caracteres la ubicación y el nombre del archivo que contiene la geometría del recinto, bien sea un archivo `.env` o `.cad`.
  - (b) `Filestem`: introducir una cadena de caracteres con la palabra con la que se desea comiencen los archivos generados en la simulación.
  - (c) `Filepath`: incluir una cadena de caracteres con el directorio donde se desea que se guarden los archivos de la simulación. Terminar la cadena con un separador.
  - (d) `int_or_ext_model`: incluir la opción 'int' o la opción 'ext' en función de si se desea un modelo interior o exterior.

- (e) `open_or_closed_model`: incluir la opción 'open' o 'closed' según se desee un modelo abierto o cerrado respectivamente.
- (f) `EDcalcmethod`: se debe introducir siempre el valor 'n' ya que esto hace que se calcule la respuesta al impulso de difracción por el método Svensson.
- (g) `specorder`: introducir el valor del orden de reflexión que se desea simular.
- (h) `difforder`: introducir el valor del orden de difracción a simular, siempre y cuando no sea mayor que el orden de reflexión y si este último es mayor o igual que 3 debe tener un valor 0.
- (i) `SHOWTEXT`: un número de 1 a 4, donde a mayor valor mayor cantidad de texto es mostrado en la ventana de comandos de Matlab durante la simulación.
- (j) `calcirs`: debe tener un valor 1 si se desea que se calcule las respuestas al impulso, y un valor 0 en caso contrario.
- (k) `Rstart`, `FSAMP`, `CAIR`, `RHOAIR`: explicadas en apartado [B.2.2.3](#).
- (l) `separacion_point_edges`: por último, introducir el valor en metros de la distancia a la que se desea representar los rayos dentro de un borde, partiendo del centro del mismo.

Esta función devolverá una estructura denominada `infos` con los valores necesarios para realizar simulaciones modificando las posiciones de las fuentes o receptores, sin necesidad de llamar a esta función nuevamente, a menos que se quiera modificar algún parámetro de los que aquí se recogen.

2. Se deberá llamar a la función `getpaths()` indicando en el primer parámetro una matriz en la que cada fila contendrá las posiciones de las fuentes, en el segundo una matriz en la que cada fila contendrá las posiciones de los receptores, y en tercer lugar la estructura obtenida con la ejecución de la función `InitSimulation`.

Esta función devuelve una estructura de nombre `raytype` donde cada fila representa una trayectoria calculada y las columnas el orden de la trayectoria indicando para cada orden el tipo de fenómeno producido. Con un 115 se representa una reflexión especular, un 110 una difracción en borde y un 102 un rayo directo. En la Tabla [B.1](#) se representan los distintos patrones que pueden representarse en esta estructura.

Rayo directo	102	0	0	0
Rayo especular orden 1	115	0	0	0
Rayo especular orden 2	115	115	0	0
Rayo especular orden 3	115	115	115	0
Rayo especular orden 4	115	115	115	115
Especular y difractado	115	100	0	0
Difractado y especular	100	115	0	0
Difracción orden 1	100	0	0	0
Difracción orden 2	100	100	0	0

Tabla B.1: Ejemplos de valores en matriz `raytype`

La función `getpaths` también devuelve una estructura de nombre `paths` en la que cada fila representa una trayectoria y las columnas se representan las coordenadas por las que va pasando dicha trayectoria. En la Tabla [B.2](#) se muestra el esquema de como se representan los datos en cada una de las matrices de la estructura.

Esta función devuelve dos estructuras para dentro de las mismas poder diferenciar en el caso de que se haya introducido más de una fuente y/o receptor.

raytype	$x, y, z$	$x, y, z$	$x, y, z$	$x, y, z$	$x, y, z$	$x, y, z$
Rayo directo	fuelle	receptor				
Rayo especular orden 1	fuelle	punto 1	receptor			
Rayo especular orden 2	fuelle	punto 1	punto 2	receptor		
Rayo especular orden 3	fuelle	punto 1	punto 2	punto 3	receptor	
Rayo especular orden 4	fuelle	punto 1	punto 2	punto 3	punto 4	receptor
Especular y difractado	fuelle	punto 1	punto 2	receptor		
Difractado y especular	fuelle	punto 1	punto 2	receptor		
Difracción orden 2	fuelle	punto 1	punto 2	receptor		

Tabla B.2: Ejemplos de valores en matriz paths



Universidad de Alcalá  
Escuela Politécnica Superior



ESCUELA POLITECNICA  
SUPERIOR



Universidad  
de Alcalá